DB2 Universal Database for z/OS

Version 8

Messages

Information Management software

GC18-9602-03

DB2 Universal Database for z/OS

Version 8

Messages

Information Management software

GC18-9602-03

Note

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 685.

Fourth Edition, Softcopy Only (February 2008)

This edition applies to Version 8 of IBM DB2 Universal Database for z/OS (DB2 UDB for z/OS), product number 5625-DB2, and to any subsequent releases until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

This softcopy version is based on the printed edition of the book and includes the changes indicated in the printed version by vertical bars. Additional changes made to this softcopy version of the book since the hardcopy book was published are indicated by the hash (#) symbol in the left-hand margin. Editorial changes that have no technical significance are not noted.

This and other books in the DB2 UDB for z/OS library are periodically updated with technical changes. These updates are made available to licensees of the product on CD-ROM and on the Web (currently at www.ibm.com/software/data/db2/zos/library.html). Check these resources to ensure that you are using the most current information.

© Copyright International Business Machines Corporation 1983, 2008. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Part 1. Introduction . . . 1

Chapter 1. About this book . . . 3
Who should read this book . . . 3
How to use this material . . . 3
Message type codes . . . 4
DB2 message severity codes . . . 5
DB2 abend completion codes (X'04E' and X'04F) . . . 5
DB2 diagnostic information . . . 6
MVS abends resulting from DB2 operation . . . 7
Summary of changes . . . 7
Using LookAt to obtain message explanations . . . 7
Terminology and citations . . . 7
Accessibility . . . 8
How to send your comments . . . 8

Part 2. DB2 messages . . . 9

Chapter 2. DSNA... messages . . . 11

Chapter 3. DSNB... messages . . . 27

Chapter 4. DSNE... messages . . . 71

Chapter 5. DSNF... messages . . . 97

Chapter 6. DSNG... messages . . . 99

Chapter 7. DSNH... messages . . . 101
Message line numbers for DSNH... messages . . . 101
Message severity codes for DSNH... messages . . . 101
Source tokens in DSNH... messages . . . 102
Messages in DSNH . . . 102

Chapter 8. DSNI... messages . . . 145

Chapter 9. DSNJ... messages . . . 155

Chapter 10. DSNL... messages . . . 201

Chapter 11. DSNM... messages . . . 247

Chapter 12. DSNN... message . . . 249

Chapter 13. DSNP... messages . . . 251

Chapter 14. DSNQ... messages . . . 265

Chapter 15. DSNR... messages . . . 269

© Copyright IBM Corp. 1983, 2008

Chapter 16. DSNS... messages 281

Chapter 17. DSNT... messages 283

Chapter 18. DSNU... messages 357

Chapter 19. DSNV... messages 519

Chapter 20. DSNW... messages 539

Chapter 21. DSNX... messages 553

Chapter 22. DSNY... messages 575

Chapter 23. DSNZ... messages 577

Chapter 24. DSN1... messages 579

Chapter 25. DSN2... messages 609

Chapter 26. DSN3... messages 619

Chapter 27. DSN7... messages 631

Chapter 28. DSN8... messages 639

Chapter 29. DSN9... messages 649

Part 3. Appendixes 657

Appendix A. Problem determination 659
Diagnostic items 659
Resource types 662
ABEND codes associated with DSN1COPY misuse 665
Explanations 667

Appendix B. New, changed, and deleted messages 669
New messages 669
Changed messages 677
Deleted messages 682

Notices 685
Programming interface information 686
Trademarks 687

Glossary 689

Bibliography 723

Part 1. Introduction

This information helps you to understand the messages that you could receive while working with DB2.

Unless it is stated otherwise, this information assumes that DB2® is running in new-function mode (as opposed to compatibility mode or enabling-new-function mode).

This section contains specific information about messages and a general overview of the DB2 library.

This section contains information about the following topics:

- "Who should read this book" on page 3
- "How to use this material" on page 3
- "Summary of changes" on page 7
- "Using LookAt to obtain message explanations" on page 7
- "Terminology and citations" on page 7
- "Accessibility" on page 8

© Copyright IBM Corp. 1983, 2008

Chapter 1. About this book

Throughout the messages information, "DB2" means the program product DB2 Universal Database for z/OS (DB2 UDB for z/OS). Other DB2 products are given their complete names or abbreviations.

Important

In this version of DB2 UDB for z/OS, the DB2 Utilities Suite is available as an optional product. You must separately order and purchase a license to such utilities, and discussion of those utility functions in this publication is not intended to otherwise imply that you have a license to them. See Part 1 of *DB2 Utility Guide and Reference* for packaging details.

The DB2 Utilities Suite is designed to work with the DFSORT program, which you are licensed to use in support of the DB2 utilities even if you do not otherwise license DFSORT for general use. If your primary sort product is not DFSORT, consider the following informational APARs mandatory reading:

- II14047/II14213: USE OF DFSORT BY DB2 UTILITIES
- II13495: HOW DFSORT TAKES ADVANTAGE OF 64-BIT REAL ARCHITECTURE

These informational APARs are periodically updated.

Who should read this book

DB2 Messages lists messages issued by DB2, with explanations and suggested responses. This information is intended for users and system programmers who might encounter DB2 messages.

How to use this material

DB2 Messages consists of the following topics:

- Part 1, "Introduction," on page 1 explains the different types of messages and codes that DB2 generates.
- Part 2, "DB2 messages," on page 9 lists the various DB2 messages.
- Appendix A, "Problem determination," on page 659 lists the diagnostic items and resource types you can use to resolve problems.
- Appendix B, "New, changed, and deleted messages," on page 669 lists the new, changed, and deleted messages.
- "Notices" on page 685 lists contact information and legal notices about this product

The introduction provides general information about DB2 messages. You will be referred to it from appropriate places in the documentation. However, it is recommended that you become familiar with this material ahead of time.

The DB2 messages are listed in order of their subcomponent identifier and further by their numeric identifier. All DB2 message identifiers are eight to ten characters long. The first three characters are DSN, which is the DB2 component prefix. The

© Copyright IBM Corp. 1983, 2008

fourth character is the *subcomponent identifier* which identifies the subcomponent of DB2 that issued the message. Each subcomponent has an associated *hexadecimal identifier* (hex ID) which is the hexadecimal representation of the subcomponent identifier. The *numeric identifier* is three to five characters long and begins at the fifth character in the message. The numeric identifier is unique within the subcomponent. The last character of the message identifier is the *message type code*. For DB2 messages in general, the I type code is used. However, some older messages use other type codes. The type code does not necessarily reflect the severity of the message. See the Response sections for a message to determine the required actions associated with that message.

Message identifiers have the following format:

```
DSNB209I
```

where:

DSN is the DB2 subsystem prefix name.

B is the subcomponent identifier; in this example, B identifies the DB2 buffer manager, and the hexadecimal ID for this subcomponent is X'C2'.

209 is the numeric identifier.

I indicates that this message is for information only.

Accompanying each message is the following information, when applicable:

Explanation: This section tells what the message means, why it occurred, and what caused it.

System action: This part tells what is happening as a result of the condition causing the message. If this information is not shown, no system action is taken.

User response: If a response by the user is necessary, this section tells what the pertinent responses are and their effect. If this information is not shown, no user response is required.

Operator response: If an operator response is necessary, this section tells what the pertinent responses are and their effect. If this information is not shown, no operator response is required.

System programmer response: If a response by the system programmer is needed, this part tells what the pertinent responses are and their effect. If this information is not shown, no system programmer response is required.

Programmer response: If a programmer response is necessary, this part tells what the pertinent responses are and their effect. If this information is not shown, no programmer response is required.

Problem determination: This section lists the actions that can be performed to obtain adequate data for support personnel to diagnose the problem. If this information is not shown, no problem determination is required.

Message type codes

The last character of a message identifier is the *type code*. Four type codes are used, as shown in the following list:

A	Immediate action	Action is required immediately. The associated task does not continue until the requested action is taken.
D	Immediate decision	Decision or action is required immediately. The associated task does not continue until the requested decision is made or action is taken.

E	Eventual action	Action is required at some point; however, the associated task continues. All RC=8 error messages that are not host related (SOFT errors) have the E changed to a W in precompiler output when SQL(ALL) is in effect. All RC=8 errors (HARD errors) look the same as they have in the past.
I	Information only	No action is required.

DB2 message severity codes

Messages issued by the precompiler subcomponent of DB2, as well as some messages issued by the service controller and utility subcomponents of DB2, provide severity codes. The specific meanings of the severity codes differ slightly for each of the subcomponents. In general, the higher the severity level, the more significant the situation.

Severity codes for these subcomponents are described at the beginning of the individual subcomponent sections in Part 2, “DB2 messages,” on page 9. They are also included in the individual messages, where appropriate.

DB2 abend completion codes (X'04E' and X'04F)

DB2 uses two system abend completion codes: X'04E' and X'04F'.

X'04E' and X'04F' identify:
- Internal errors encountered during DB2 operation
- Diagnostic information for problem determination
- Subsystem actions initiated by the DB2 subcomponent involved in the error

X'04E'
An X'04E' abend completion code indicates that DB2 detected an internal error and abended a DB2 internal task (TCB) or a user-connected task. Errors associated with an X'04E' abend completion code might be preceded by an MVS system code or by internal errors.

The diagnostic material generated by the X'04E' abend must be examined to determine the source of the error that actually resulted in a subsequent task or subsystem termination.

X'04F'
An X'04F' abend completion code indicates that DB2 detected a severe error and abended the entire DB2 subsystem. When an X'04F' is issued, DB2 determined that continued operation could result in the loss of data integrity. Errors associated with an X'04F' abend completion code might be preceded by an MVS system error or by one or more DB2 X'04E' abend completion codes.

Table 1 on page 6 summarizes the diagnostic information and subsystem actions available to DB2 when these abend completion codes are issued. Some pieces of this information are not relevant to some error situations and will not help in problem diagnosis. In these cases, DB2 does not produce any diagnostic information that is unnecessary. The information that DB2 produces for a given error depends upon the specific problem. The MVS services that provide diagnostic information are discussed in “DB2 diagnostic information” on page 6

Table 1. Abend Completion Codes

	X'04E'	X'04F'
Explanation	• Error during DB2 normal operation	• Severe error; continued operation may jeopardize data integrity
System action	• Internal DB2 task is abended • Connected user task is abended	• The entire DB2 subsystem is abended • User task with an active DB2 connection may be abnormally terminated with an X'04F' • Possible MEMTERM (memory termination) of connected allied address space
Diagnostic information	• SVC dump • SYS1.LOGREC entries • VRA data entries	• SYS1.LOGREC entries • VRA data entries
Associated reason codes	• DB2 abend reason code • Associated MVS system codes	• Subsystem termination reason code • MVS system completion codes and X'04E' codes that precede the X'04F' abend
Location of accompanying codes	• SVC dump title • Message DSNW050I • Register 15 of SDWA section "General Purpose Registers at Time of Error" • SYS1.LOGREC entries • VRA data entries	• SYS1.LOGREC entries • VRA data entries • Message DSNV086E, which is sent to MVS system operator

DB2 diagnostic information

DB2's functional recovery routines use MVS services to provide diagnostic information for assistance in problem determination.

The following MVS services provide diagnostic information:

- SVC dumps

  The DB2 abend completion code X'04E' uses the MVS SDUMP service to create SVC dumps. The content and storage areas associated with these dumps vary depending on the specific error and the state of the DB2 subsystem at the time the error.

- SYS1.LOGREC entries

  Entries are requested in the SYS1.LOGREC data set at the time of the error using the MVS SETRP service. The following are also recorded in SYS1.LOGREC:
  - Subsystem abnormal terminations
  - Secondary abends occurring in a recovery routine
  - Recording requests from recovery routines percolated to by the recovery termination manager.

- Variable recording area (VRA) data

  Data entries are added to the VRA of the SDWA by using an MVS VRA defined key. VRA data includes a series of diagnostic data entries common to both DB2 abend completion codes. Additional information is provided by the invoking subcomponent recovery routine during initial error processing or recovery termination manager percolation.

MVS abends resulting from DB2 operation

During DB2 operation, an abend can occur with an MVS system completion code. If you receive an MVS abend during DB2 operation, refer to the appropriate MVS publication.

Summary of changes

This edition adds and modifies messages to support Version 8 of DB2 UDB for z/OS, and it incorporates technical corrections and changes made to previously published information. This book, along with *DB2 Codes* (GC18-9603), replaces the Version 8 *DB2 Messages and Codes* (GC18-7422).

For specific information about new, changed and deleted messages Version 8, see Appendix B, “New, changed, and deleted messages,” on page 669.

Using LookAt to obtain message explanations

LookAt is an online facility that displays explanations for most messages that you encounter, and for some system abends and codes.

You can use LookAt:

- From the Internet at:

  www.ibm.com/eserver/zseries/zos/bkserv/lookat/
- From anywhere in z/OS where you can access a TSO/E command line (for example, a TSO/E prompt, ISPF, or z/OS UNIX System Services running OMVS). To use LookAt as a TSO/E command, LookAt must be installed on your host system. Obtain the LookAt code for TSO/E from a disk on your z/OS Collection (SK3T-4269) or from the LookAt ftp site:

  ftp.software.ibm.com/ps/products/ibmreader/tools/lookat/ZOS/
- From your Palm VIIx personal data assistant (PDA). To use LookAt from your Palm VIIx PDA, LookAt must be installed on the PDA. Obtain the LookAt code from a disk on your z/OS Collection (SK3T-4269) or from the LookAt ftp site:

  ftp.software.ibm.com/ps/products/ibmreader/tools/lookat/PALM/

Terminology and citations

In this information, DB2 Universal Database™ for z/OS® is referred to as "DB2 UDB for z/OS." In cases where the context makes the meaning clear, DB2 UDB for z/OS is referred to as "DB2." When this information refers to titles of books in this library, a short title is used. (For example, "See *DB2 SQL Reference*" is a citation to *IBM® DB2 Universal Database for z/OS SQL Reference*.)

When referring to a DB2 product other than DB2 UDB for z/OS, this information uses the product’s full name to avoid ambiguity.

The following terms are used as indicated:

DB2 Represents either the DB2 licensed program or a particular DB2 subsystem.

OMEGAMON
Refers to any of the following products:
- IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS
- IBM Tivoli OMEGAMON XE for DB2 Performance Monitor on z/OS
- IBM DB2 Performance Expert for Multiplatforms and Workgroups
- IBM DB2 Buffer Pool Analyzer for z/OS

C, C++, and C language
Represent the C or C++ programming language.

CICS® Represents CICS Transaction Server for z/OS or CICS Transaction Server for OS/390®.

IMS™ Represents the IMS Database Manager or IMS Transaction Manager.

MVS™ Represents the MVS element of the z/OS operating system, which is equivalent to the Base Control Program (BCP) component of the z/OS operating system.

RACF®
Represents the functions that are provided by the RACF component of the z/OS Security Server.

Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products. The major accessibility features in z/OS products, including DB2 UDB for z/OS, enable users to:

- Use assistive technologies such as screen reader and screen magnifier software
- Operate specific or equivalent features by using only a keyboard
- Customize display attributes such as color, contrast, and font size

Assistive technology products, such as screen readers, function with the DB2 UDB for z/OS user interfaces. Consult the documentation for the assistive technology products for specific information when you use assistive technology to access these interfaces.

Online documentation for Version 8 of DB2 UDB for z/OS is available in the Information management software for z/OS solutions information center, which is an accessible format when used with assistive technologies such as screen reader or screen magnifier software. The Information management software for z/OS solutions information center is available at the following Web site:
http://publib.boulder.ibm.com/infocenter/dzichelp

How to send your comments

Your feedback helps IBM to provide quality information. Please send any comments that you have about this book or other DB2 UDB for z/OS documentation. You can use the following methods to provide comments:

- Send your comments by e-mail to db2zinfo@us.ibm.com and include the name of the product, the version number of the product, and the number of the book. If you are commenting on specific text, please list the location of the text (for example, a chapter and section title or a help topic title).
- You can send comments from the Web. Visit the library Web site at:

  www.ibm.com/software/db2zos/library.html

  This Web site has a an online reader comment form that you can use to send comments.
- You can also send comments by using the feedback link at the footer of each page in the Information Management Software for z/OS Solutions Information Center at http://publib.boulder.ibm.com/infocenter/db2zhelp.

Part 2. DB2 messages

Refer to Part 1, “Introduction,” on page 1 for a description of message types and message numbering conventions.

Unless the message is issued by a subcomponent operating outside the DB2 environment, messages are preceded by a *command prefix* that is defined during system installation. The command prefix is not shown in the message text.

In the TSO attachment facility, the user can control whether message IDs are displayed. To see them, type 'PROFILE MSGID'. To suppress them, type 'PROFILE NOMSGID'.

This section contains information about the following topics:

- Chapter 2, “DSNA... messages,” on page 11
- Chapter 3, “DSNB... messages,” on page 27
- Chapter 4, “DSNE... messages,” on page 71
- Chapter 5, “DSNF... messages,” on page 97
- Chapter 6, “DSNG... messages,” on page 99
- Chapter 7, “DSNH... messages,” on page 101
- Chapter 8, “DSNI... messages,” on page 145
- Chapter 9, “DSNJ... messages,” on page 155
- Chapter 10, “DSNL... messages,” on page 201
- Chapter 11, “DSNM... messages,” on page 247
- Chapter 12, “DSNN... message,” on page 249
- Chapter 13, “DSNP... messages,” on page 251
- Chapter 14, “DSNQ... messages,” on page 265
- Chapter 15, “DSNR... messages,” on page 269
- Chapter 16, “DSNS... messages,” on page 281
- Chapter 17, “DSNT... messages,” on page 283
- Chapter 18, “DSNU... messages,” on page 357
- Chapter 19, “DSNV... messages,” on page 519
- Chapter 20, “DSNW... messages,” on page 539
- Chapter 21, “DSNX... messages,” on page 553
- Chapter 22, “DSNY... messages,” on page 575
- Chapter 23, “DSNZ... messages,” on page 577
- Chapter 24, “DSN1... messages,” on page 579
- Chapter 25, “DSN2... messages,” on page 609
- Chapter 26, “DSN3... messages,” on page 619
- Chapter 27, “DSN7... messages,” on page 631
- Chapter 28, “DSN8... messages,” on page 639
- Chapter 29, “DSN9... messages,” on page 649

© Copyright IBM Corp. 1983, 2008

Chapter 2. DSNA... messages

This section contains two types of messages:

1. Messages that are written to a trace file. These are seen only if the trace is turned on (that is, ddname DSNTRACE was allocated to SYSOUT or the TSO terminal).
2. Messages that are returned by a DB2 stored procedure. These are seen only as a parameter that is returned by the stored procedure.

DSNA... messages share a basic structure. Here are the message elements, listed in order:

1. Message number (DSNAnnna).
2. The name of the CSECT responsible for writing the message.
3. TCB address of the task currently executing.
4. A series of tokens and token-explanations unique to each message.

Note: The intent of these messages is to aid IBM development and the IBM Support Center. They contain information that might not be meaningful except to IBM.

DSNA002E DSNACA00 TCB=address SYSTEM ERROR - FOUND UNEXPECTED THREAD

Explanation: The DB2 call attachment facility control block, the CAB, indicated that a thread was present when none should have been present. This is a CAF system error.

This message is issued by the following CSECT: DSNACA00

System action: This message immediately precedes an 04E abend, with reason code 00C10002.

System programmer response: Collect a SYSUDUMP and DSNTRACE trace stream.

Programmer response: Refer the problem to your system programmer.

Problem determination: Contact your IBM Support Center.

DSNA201E DSNACA00 TCB=address CANNOT USE 'CONNECT', ALREADY CONNECTED TO DB2

Explanation: A CONNECT request was received but a connection to DB2 already existed for the TCB identified by 'address' in the message. This is a CAF error (application program error). The message is accompanied by return code (R15) 200 and code (R0) 00C10201.

System action: No new connection is made to DB2 and an error return and reason code are issued. The DB2 call attachment facility is ready for more input.

System programmer response: Refer the problem to the programmer using the DB2 call attachment facility.

Programmer response: You must DISCONNECT the previous connection before you can re-CONNECT. Correct the logic in your code.

Problem determination: See a DSNTRACE trace stream to understand the context of this message.

DSNA202E DSNACA00 TCB=address CANNOT USE 'OPEN', PRIOR OPEN OR SQL ALREADY CONNECTED APPLICATION TO DB2

Explanation: An OPEN request was received but a DB2 thread had already been created for the specified TCB. This is a user error. The message is accompanied by return code (R15) 200 and code (R0) 00C10202.

System action: No new DB2 thread is created and an error return and reason code are issued. The DB2 call attachment facility is ready for more input.

System programmer response: Refer the problem to the programmer using the DB2 call attachment facility.

Programmer response: Your application program had already (explicitly or implicitly) established an OPEN level connection to DB2. You cannot do this again until the original OPEN is terminated. 'Explicit' OPENs occur when an application issues the call attachment facility OPEN request. 'Implicit' OPENs occur when an application issues SQL statements before a DB2

© Copyright IBM Corp. 1983, 2008

connection was established. Correct the logic in your application program.

Problem determination: See a DSNTRACE trace stream to understand the context of this message.

DSNA203E DSNACA00 TCB=address CANNOT USE 'CLOSE', THERE WAS NO 'OPEN' IN EFFECT

Explanation: A CLOSE request was received but no OPEN level connection (implicit or explicit) existed for this task. This is a user error. The message is accompanied by return code (R15) 200 and code (R0) 00C10203.

System action: No CLOSE is performed. The DB2 call attachment facility is ready for more input.

System programmer response: Refer the problem to the programmer using the DB2 call attachment facility.

Programmer response: Correct the logic in your application. Do not issue CLOSE unless an explicit OPEN has been performed. 'Explicit' OPENs occur when an application issues the call attachment facility OPEN request.

Problem determination: See a DSNTRACE trace stream to understand the context of this message.

DSNA204E DSNACA00 TCB=address CANNOT USE 'DISCONNECT', THERE WAS NO 'CONNECT' IN EFFECT

Explanation: A DISCONNECT request was received but no DB2 connection had been established. This is a user error. The message is accompanied by return code (R15) 200 and code (R0) 00C10204.

System action: No DISCONNECT has been performed. The DB2 call attachment facility is ready for more input.

System programmer response: Refer the problem to the programmer using the DB2 call attachment facility.

Programmer response: Correct the logic in your application. Do not issue DISCONNECT unless a CONNECT has been performed.

Problem determination: See a DSNTRACE trace stream to understand the context of this message.

DSNA205E DSNACA00 TCB=address CANNOT USE 'TRANSLATE', THERE WAS NO CONNECTION TO DB2 OR NOTHING TO TRANSLATE

Explanation: A TRANSLATE request was received but no data was available for translation. You must have valid DB2 call attachment facility control structures before you can make a TRANSLATE request. Control structures are built when you issue explicit or implicit connection requests. This is a user error. The message is accompanied by return code (R15) 200 and code (R0) 00C10205.

System action: No TRANSLATE is performed. The DB2 call attachment facility is ready for more input.

System programmer response: Refer the problem to the programmer using the DB2 call attachment facility.

Programmer response: Correct the logic in your application. Do not issue TRANSLATE unless a DB2 connection has been attempted.

Problem determination: See a DSNTRACE trace stream to understand the context of this message.

DSNA206E DSNACA00 TCB=address WRONG NUMBER OF PARMS OR VL BIT OFF

Explanation: An erroneous DB2 call attachment facility request was received. The request had too many or too few parameters and could not be processed. The message is accompanied by return code (R15) 200 and code (R0) 00C10206.

System action: The previous request was not processed. The DB2 call attachment facility is ready for more input.

System programmer response: Refer the problem to the programmer using the DB2 call attachment facility.

Programmer response: Correct the logic in your application. Be sure that any parameter list passed to the DB2 call attachment facility has the correct number of parameters for the individual call, and that the high order (VL) bit is turned on to end the parameter list.

Problem determination: See a DSNTRACE trace stream to understand the context of this message.

DSNA207E DSNACA00 TCB=address DID NOT RECOGNIZE ACTION PARAMETER - parm

Explanation: An unrecognized action request type or parameter was received by the DB2 call attachment facility and could not be processed. The allowable action request types are CONNECT, OPEN, CLOSE, DISCONNECT, and TRANSLATE. For allowable parameters, refer to the CAF section in *DB2 Application Programming and SQL Guide*. This message is accompanied by return code (R15) 200 and code (R0) 00C10207.

System action: Control returns to the caller. Your request was not processed. The DB2 call attachment facility is ready for more input.

System programmer response: Refer the problem to the programmer using the DB2 call attachment facility.

Programmer response: Correct the logic in your application. Check the spelling of the action parameter type or keyword on the previous request passed to the DB2 call attachment facility. The action parameter type

you passed is listed in the parm field of the DSNA800I trace message. If the error is not the action parameter, the parameter in error is listed in the parm field of this message.

Problem determination: See a DSNTRACE trace stream to understand the context of this message.

DSNA208E DSNACA00 TCB=address CANNOT HAVE 2 SSIDS (ssid1 AND ssid2) FROM 1 TCB

Explanation: An OPEN request was received specifying a different subsystem identifier (ssid2) than the previous CONNECT request (ssid1). Your task can only be connected to a single DB2 subsystem. The message is accompanied by return code (R15) 200 and code (R0) 00C10208.

System action: Control returns to the caller. Your request was not processed. The DB2 call attachment facility is ready for more input.

System programmer response: Refer the problem to the programmer using the DB2 call attachment facility.

Programmer response: Specify the correct DB2 subsystem identifier on both the OPEN and CONNECT requests.

Problem determination: See a DSNTRACE trace stream to understand the context of this message.

DSNA209E DSNACA00 A STORED PROCEDURE ISSUED AN EXPLICIT CALL TO DSNALI

Explanation: A stored procedure issued an explicit call to DSNALI. This is not allowed. Return code 200 in register 15 and code 00C10209 in register 0 accompany this message.

System action: Control returns to the caller. The request was not processed. The DB2 call attachment facility is ready for more input.

System programmer response: Refer the problem to the programmer using the DB2 call attachment facility.

Programmer response: Remove explicit calls to DSNALI from the application program.

Problem determination: See a DSNTRACE trace job stream to understand the context of this message.

DSNA301I THE VALUE *parm-value* FOR PARAMETER *parm-name* IS NOT VALID

Explanation: The stored procedure DSNACICS could not run because one of the input parameters contains an incorrect value.

Severity: 12

System action: The stored procedure DSNACICS terminates without calling the CICS transaction.

Programmer response: Correct the input parameter value specified on the SQL CALL statement.

System programmer response: Ensure that the user exit DSNACICX sets the parameter to a valid value.

DSNA302I THE REQUIRED INPUT PARAMETER *parm-name* CONTAINS NULLS

Explanation: The stored procedure DSNACICS could not run because one or more of the required input parameters contains a null value.

Severity: 12

System action: The stored procedure DSNACICS terminates without calling the CICS transaction.

Programmer response: Correct the input parameter value in the SQL CALL statement, or ensure that the user exit DSNACICX sets the parameter to a non-null value.

System programmer response: Ensure that the user exit DSNACICX sets the parameter to a non-null value.

DSNA303I THE PARAMETER NETNAME WAS NOT SPECIFIED WHEN THE CONNECT TYPE OF SPECIFIC WAS SPECIFIED

Explanation: The input parameter NETNAME must contain a non-null value when the CONNECT TYPE parameter is set to SPECIFIC.

Severity: 12

System action: The stored procedure DSNACICS terminates without calling the CICS transaction.

Programmer response: Correct the input parameter value in the SQL CALL statement, or ensure the the user exit DSNACICX sets the parameter to a non-null value.

System programmer response: Ensure that the user exit DSNACICX sets the parameter to a non-null value.

DSNA304I THE USER EXIT DSNACICX RETURNED RETURN CODE = X'hhhhhhhh' AND THE FOLLOWING ERROR MESSAGE:

Explanation: The user exit DSNACICX returned a non-zero return code and a message to stored procedure DSNACICS. The message that DSNACICX returns immediately follows message DSNA304I.

Severity: 12

System action: The stored procedure DSNACICS terminates without calling the CICS transaction.

Programmer response: Contact the system programmer.

System programmer response: Correct the cause of

the error message that the user exit generated.

DSNA305I THE CICS EXCI *func* REQUEST FAILED. CICS RESPONSE CODE = X'hhhh' CICS REASON CODE = X'hhhhhhhh'

Explanation: DSNACICS uses the CICS EXCI Call interface to request the execution of a CICS server program. CICS EXCI interface call *func* resulted in an unexpected CICS response code and reason code. Possible values of *func* are:

Initialize_User Initializes the EXCI user environment.

Allocate_Pipe Allocates a session to a CICS region.

Open_Pipe Connects an allocated pipe to a receive session of the CICS region.

DPL_Request Issues a request to execute the CICS server program.

Close_Pipe Disconnects an open pipe from a CICS region.

Deallocate_Pipe Deallocates a pipe from a CICS region.

The failure can occur because the EXCI interface is not properly enabled for the CICS subsystem. For explanations of the CICS response code and reason code, see *CICS Transaction Server for z/OS External Interfaces Guide*.

Severity: 12

System action: The stored procedure DSNACICS terminates without calling the CICS transaction.

Programmer response: Contact the system programmer.

System programmer response: Ensure that the EXCI interface is enabled to the CICS region that the stored procedure calls. If you suspect an error in the DB2 stored procedure, refer to Part 3 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Add a DSNDUMP DD statement to the JCL startup procedure for the stored procedure address space in which DSNACICS runs. Then rerun the application that calls DSNACICS. This causes DB2 to generate an SVC dump whenever it issues an error message for DSNACICS.

If you suspect an error in the DB2 stored procedure, refer to Part 3 of *DB2 Diagnosis Guide and Reference*. Collect the following diagnostic items listed in Appendix A, “Problem determination,” on page 659: 1, 2, 22.

DSNA306I THE CICS SERVER PROGRAM *program* ABENDED. CICS ABEND CODE=C'cccc'

Explanation: The CICS server program that DSNACICS called abnormally terminated. CICS returns a four-character abend code to indicate the error. These error codes are described in *CICS Transaction Server for z/OS Problem Determination Guide*.

Severity: 12

System action: Stored procedure DSNACICS processes the next request. CICS does a backout to the last syncpoint and forces the SYNCONRETURN option.

Programmer response: Correct the error in the server program.

System programmer response: Correct the cause of the error in the server program. Ensure that the mirror transaction is still enabled after this error. The mirror transaction must be enabled so that subsequent requests to run a server program are accepted.

DSNA307I OS/390 *token-service* REQUEST FAILED. SYSTEM RETURN CODE=X'hhhh'

Explanation: DSNACICS uses the OS/390 token services to save task-level information. The OS/390 service returned an unexpected return code.

token-service is the token service function that DSNACICS requested. Possible values are:

Create_token Create a token using the IEANTCR service.

Retrieve_token Retrieve a token using the IEANTRT service.

See the description of IEANTCR or IEANTRT in *z/OS MVS Programming: Assembler Services Reference, Volumes 1 and 2* for a full description of the return code.

Severity: 12

System action: Stored procedure DSNACICS processes the next request.

Programmer response: Contact the system programmer.

System programmer response: Add a DSNDUMP DD statement to the JCL startup procedure for the stored procedure address space in which DSNACICS runs. Then rerun the application that calls DSNACICS. This causes DB2 to generate an SVC dump whenever it issues an error message for DSNACICS.

If you suspect an error in the DB2 stored procedure, refer to Part 3 of *DB2 Diagnosis Guide and Reference*. Collect the following diagnostic items listed in Appendix A, “Problem determination,” on page 659: 1, 2, 22.

DSNA308I THE CICS EXCI DPL_REQUEST REQUEST FAILED. CICS RESPONSE CODE=0. DPL RETAREA CONTAINS FIELD1=X'hhhhhhhh', FIELD2=X'hhhhhhhh', FIELD3=C'cccc'

Explanation: DSNACICS uses the CICS EXCI Call interface to request the execution of a CICS server program. The DPL_REQUEST command returned an unexpected value in the DPL_RETAREA. See *CICS Transaction Server for z/OS External Interfaces Guide* and *CICS Application Programming Guide* for a full description of the meanings of FIELD1, FIELD2, and FIELD3.

Severity: 12

System action: Stored procedure DSNACICS processes the next request.

Programmer response: Correct any errors in the server program. Contact the system programmer.

System programmer response: Add a DSNDUMP DD statement to the JCL startup procedure for the stored procedure address space in which DSNACICS runs. Then rerun the application that calls DSNACICS. This causes DB2 to generate an SVC dump whenever it issues an error message for DSNACICS.

If you suspect an error in the DB2 stored procedure, refer to Part 3 of *DB2 Diagnosis Guide and Reference*. Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 22.

DSNA309I UNABLE TO LOAD USER EXIT DSNACICX. SYSTEM REASON CODE=X'hhhh' SYSTEM COMPLETION CODE=X'hhhh'

Explanation: DSNACICS was unable to load the user exit routine DSNACICX.

Severity: 12

System action: The stored procedure DSNACICS does not call the CICS transaction. DSNACICS processes the next request.

Programmer response: Contact the system programmer.

System programmer response: Ensure that the STEPLIB statement in the startup procedure for the stored procedure address space specifies the library that contains DSNACICX. A default user exit is shipped in the SDSNLOAD DB2 library.

Add a DSNDUMP DD statement to the JCL startup procedure for the stored procedure address space in which DSNACICS runs. Then rerun the application that calls DSNACICS. This causes DB2 to generate an SVC dump whenever it issues an error message for DSNACICS.

If you suspect an error in the DB2 stored procedure, refer to Part 3 of *DB2 Diagnosis Guide and Reference*. Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 22.

DSNA310I STORED PROCEDURE TASK CANCELLED BY OPERATOR

Explanation: The task that called the DSNACICS stored procedure was cancelled by the operator.

Severity: 12

System action: DSNACICS terminates immediately. The CICS server program might have been called.

Programmer response: Contact the system programmer.

System programmer response: Determine why the operator cancelled the stored procedure task.

DSNA311I DSNACICS ABNORMALLY TERMINATED

Explanation: The task that called the DSNACICS stored procedure abnormally terminated.

Severity: 12

System action: DSNACICS terminates immediately. The CICS server program might have been called. An SVC dump or a SYSMDUMP is generated for this error. If a CICS service encountered an error, a SYSMDUMP might be generated. For an abend in the stored procedure, an SVC dump is generated.

Programmer response: Contact the system programmer.

System programmer response: Review the JES messages for the stored procedure address space in which DSNACICS runs. Look for CICS or system messages that describe the source of the problem.

If you suspect an error in the DB2 stored procedure, refer to Part 3 of *DB2 Diagnosis Guide and Reference*. Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 22.

DSNA312I USER EXIT DSNACICX ABNORMALLY TERMINATED

Explanation: User exit program DSNACICX abnormally terminated.

Severity: 12

System action: DSNACICS terminates immediately. The CICS server program is not called. An SVC dump is generated for this error.

Programmer response: Contact the system programmer.

System programmer response: Determine why the user exit abended. If you suspect an error in the DB2 stored procedure, refer to Part 3 of *DB2 Diagnosis Guide and Reference*. Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 22.

DSNA313I DSNACICS TERMINATED BY DB2 CANCEL THREAD COMMAND

Explanation: The stored procedure DSNACICS terminated because the operator issued the DB2 CANCEL THREAD command.

Severity: 12

System action: DSNACICS terminates immediately. The CICS server program might have been called.

Programmer response: Contact the system programmer.

System programmer response: Determine why the operator issued the DB2 CANCEL THREAD command.

DSNA314I UNABLE TO LOAD CICS STUB ROUTINE DFHXCSTB. SYSTEM REASON CODE=X'hhhh' SYSTEM COMPLETION CODE=X'hhhh'

Explanation: DSNACICS was unable to load the CICS stub routine DFHXCSTB.

Severity: 12

System action: The stored procedure DSNACICS does not call the CICS transaction. DSNACICS processes the next request.

Programmer response: Contact the system programmer.

System programmer response: Ensure that the STEPLIB statement in the startup procedure for the stored procedure address space specifies the CICS library that contains DFHXCSTB. DFHXCSTB is in the CICS EXCI library (SDFHEXCI). This library must be APF authorized.

Add a DSNDUMP DD statement to the JCL startup procedure for the stored procedure address space in which DSNACICS runs. Then rerun the application that calls DSNACICS. This causes DB2 to generate an SVC dump whenever it issues an error message for DSNACICS.

If you suspect an error in the DB2 stored procedure, refer to Part 3 of *DB2 Diagnosis Guide and Reference*. Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 22.

DSNA315I DSNAIMS FUNCTION *func-name* HAS COMPLETED SUCCESSFULLY.

Explanation: The stored procedure, DSNAIMS, executed successfully.

System action: None.

Programmer response: None.

DSNA316I DSNAIMS INPUT PARAMETER ERROR. CALLER MUST PROVIDE INPUT FOR PARAMETER *parm-name* WITH FUNCTION *func-name*.

Explanation: A value for parameter *parm-name* is required for function *func-name*.

System action: DSNAIMS execution terminated before invoking IMS.

Programmer response: Provide an appropriate value for the parameter, and re-run the function.

DSNA317I DSNAIMS ERROR IN *otmaci-api* API. RC=*return-code*, RSN1=*reason-code1*, RSN2=*reason-code2*, RSN3=*reason-code3*, RSN4=*reason-code4*.

Explanation: OTMA Callable Interface API *otmaci-api* encountered an error.

System action: The DSNAIMS execution terminated after executing the specified API call.

Programmer response: Refer to the *IMS OTMA Guide and Reference* for explanations of the return and reason codes.

DSNA318I DSNAIMS ERROR IN RRS CTXRCC API. RC=*return-code*

Explanation: An error was encountered in RRS when processing the two-phase commit.

System action: DSNAIMS execution terminated before invoking IMS.

Programmer response: Refer to the *MVS Programming Resource Recovery* manual for an explanation of the return code.

DSNA319I DSNAIMS RECEIVED UNKNOWN FUNCTION *func-name*

Explanation: DSNAIMS received an unknown function name in parameter 1 of the stored procedure call.

System action: DSNAIMS terminated before invoking IMS.

Programmer response: Specify a function name known to DSNAIMS in parameter 1.

DSNA320I DSNAIMS ERROR IN CTXSWCH DURING THE SWITCH AWAY FROM PRIVATE CONTEXT. RC=*return-code*

Explanation: An error was encountered in CTXSWCH API during the switch away from private context of DSNAIMS.

System action: DSNAIMS is terminated after executing CTXSWCH.

Programmer response: Contact IBM service for the return code.

DSNA321I DSNAIMS ERROR IN CTXSWCH DURING THE SWITCH TO PRIVATE CONTEXT. RC=*return-code*

Explanation: An error was encountered in CTXSWCH API during the switch to private context of DSNAIMS.

System action: DSNAIMS is terminated after executing CTXSWCH.

Programmer response: Contact IBM service for the return code.

DSNA601I *csect-name* **THE PARAMETER** *parm-name* **IS NOT VALID, INVALID REASON CODE =** *xx*

Explanation: The parameter specified in the message is not valid. The cause of the invalid error is identified by the INVALID REASON CODE value in the message text. The specified parameter is not valid for the indicated reason:

1. Value is not an acceptable value.
2. Value is not unique.
3. Value is null.
4. Value is blank.
5. Value is not null.
6. Value is too long.
7. Named parameter is not known.
8. Named parameter is missing.

The *csect-name* in the message identifies the CSECT that issued the message.

Severity: 12 (severe)

System action: The stored procedure or the started task terminates.

Programmer response: Correct the input parameter value specified on the SQL CALL statement or on the start command.

DSNA602I *csect-name* **INCORRECT DSN SUBCOMMAND** *subcommand* **RECEIVED**

Explanation: The DSN subcommand specified is not an acceptable value. The only supported DSN subcommands are: BIND, REBIND and FREE.

Severity: 12 (severe)

System action: Processing terminates.

Programmer response: Correct the invalid DSN subcommand.

DSNA603I *csect-name* **THE COLUMN** *column-name* **IN TABLE** *table-name* **IS NOT VALID.** *row-identifier*= *id***. INVALID REASON CODE=** *xx*

Explanation: An invalid value was supplied for *column-name* column. The invalid column was in *row-identifier*= *id* row in the *table-name* table. The cause is identified by the INVALID REASON CODE value in the message text. The column is not valid for the indicated reason:

The parameter specified in the message is not valid. The cause of the invalid error is identified by the INVALID REASON CODE value in the message text.

The specified parameter is not valid for the indicated reason:

1. The value is not an acceptable value.
2. The value is invalid.
3. The value is null.

Severity: 12 (severe)

System action: Processing terminates.

User response: Specify a valid value for the specified column.

DSNA604I *csect-name* **DUPLICATE OBJECT-ID** *object-id* **IN TABLE** *table-name*

Explanation: A duplicate value was supplied for the object ID column in *table-name* table. The duplicate value was in the row with object ID *object-id*.

Severity: 12 (severe)

System action: Processing terminates.

User response: Specify a unique object-id for the object ID column in the named table.

DSNA605I *csect-name* **THERE ARE** *object-num* **OBJECTS NOT FOUND IN THE DB2 CATALOG**

Explanation: There are *object-num* objects not found in the DB2 catalog.

Severity: 12 (severe)

System action: Objects not found in the DB2 catalog are not processed.

User response: Verify the name of the objects that were not processed.

DSNA607I *csect-name* **UNABLE TO** *member-action* **MEMBER=***member* **OF DATA SET DSNAME=***dsname***, MEMBER NOT FOUND**

Explanation: The action on the library member failed because the library member was not found.

Severity: 12 (severe)

System action: Processing terminates.

User response: Specify a valid library for *member-name*.

DSNA610I *csect-name* **INSUFFICIENT RACF AUTHORITY FOR DATA SET**

Explanation: A RACF error was encountered while processing a data set.

Severity: 12 (severe)

System action: Processing of the data set terminates.

User response: Contact the system programmer, or system administrator to arrange for the required RACF or equivalent security authorization. If more information is needed, look at the messages in the console log.

DSNA611I *csect-name* **THE** *data-set-attribute* **OF THE DATA SET DSNAME=***dsname* **IS NOT** *yy*

Explanation: The data set indicated in the message cannot be processed because the data set attribute is not supported.

Severity: 12 (severe)

System action: No action is taken with the named data set.

Programmer response: Specify a data set with the correct data set attribute.

DSNA612I *csect-name* **INSUFFICIENT SPACE FOR DATA SET**

Explanation: Insufficient disk space was encountered while writing to a data set.

Severity: 12 (severe)

System action: Processing of the data set terminates.

Programmer response: Allocate more space for the data set. If more information is needed, look at the messages in the console log.

DSNA613I *csect-name* **THE DATA SET NAME DSNAME=***dsname* **MUST NOT BEGIN WITH A PATTERN MATCHING CHARACTER**

Explanation: The specified data set name begins with an '*' or '%'. This is not allowed for filtering, because it will cause a massive search of the catalog and could tie up a lot of system resources.

Severity: 12 (severe)

System action: No action is taken with the named data set.

Programmer response: Correct the invalid data set name.

DSNA614I *csect-name* **UNABLE TO OBTAIN** *storage-usage* **STORAGE, SIZE =** *size* **BYTES**

Explanation: An error occurred when the stored procedure attempted to obtain *size* bytes of storage for the *storage-usage* indicated. The *csect-name* in the message identifies the CSECT that issued the message.

Severity: 12 (severe)

System action: The stored procedure terminates.

Programmer response: Notify the system programmer before attempting to reissue the SQL CALL statement.

DSNA615I *csect-name* **UNABLE TO LOAD** *module-name* **MODULE, LOAD R15=***rc*

Explanation: An error resulted when an attempt was made to load the module indicated in the message.

Severity: 12 (severe)

System action: Processing terminates.

Programmer response: Notify the system programmer.

System programmer response: Verify that the named load module exists, and that DB2 can access it.

DSNA616I *csect-name* **CATALOG SEARCH INTERFACE ERROR, R15=***xx***, REASON CODE=***yy* **RETURN CODE=***zz*

Explanation: An error occurred when the Catalog Search Interface (CSI) was called.

Severity: 12 (severe)

System action: Processing terminates.

Programmer response: Refer to *z/OS DFSMS: Macro Instructions for Data Sets* for information on the interface's R15 reason code, and return code values.

DSNA617I *csect-name* **IFCA ERROR, IFCARC1=***vv***, IFCARC2=***ww***, IFCABNM=***xx***, IFCAGRSN=***yy***, IFCAGBNM=***zz***

Explanation: An error occurred when submitting a DB2 command through the instrumentation facility interface (IFI). IFCARC1, IFCARC2, IFCABNM IFCAGRSN, and IFCAGBNM are fields in the IFI Communication Area (IFCA).

Severity: 12 (severe)

System action: Processing terminates.

Programmer response: Refer to *DB2 Codes*, section "X'E6.....' IFI reason codes" for information on the IFCARC1, IFCARC2, and IFCAGRSN values. Consult the Instrument facility communications area (IFCA) section of the *DB2 Administration Guide* for information about IFCABNM and IFCAGBNM.

DSNA618I *csect-name* **SQL ERROR DURING SQL STATEMENT** *sql*, *object-type* = *object-name*

Explanation: An SQL error resulted when the stored procedure issued an SQL statement *sql* for the identified DB2 object.

This message can be followed by a formatted form of the SQLCA and a text message based on the SQLCODE field of the SQLCA, or it can be followed by an unformatted form of the SQLCA.

csect-name
Identifies the CSECT that issued the message

sql The SQL statement encountering the error

object-type
The type of DB2 object that the failed SQL statement is processing

object-name
The name of the DB2 object that the failed SQL statement is processing

System action: The stored procedure issuing the SQL statement terminates. On the operator's console, the administrative scheduler that issues this message cannot access its DB2 task list table. It is operating only on a single task list

Programmer response: Follow the diagnosis approach suggested for the SQL return code.

DSNA620I *csect-name* **ERROR ALLOCATING DATA SET DSNAME=***dsname***, RETURN CODE=***xx***, ERROR REASON CODE=***yy***, INFO REASON CODE=***zz***

Explanation: An error occurred when the DYNALLOC macro was invoked to dynamically allocate the named data set.

Severity: 12 (severe)

System action: Processing terminates.

User response: Refer to *z/OS MVS Diagnosis: Reference* or *z/OS MVS Programming: Authorized Assembler Services Guide* for information on the DYNALLOC return code, ERROR reason code, and INFO reason code values.

DSNA621I *csect-name* **ATTEMPT TO PERFORM RRS ATTACH FUNCTION** *funcname* **FAILED, RRS ATTACH RC=***rc***, RETCODE=***retcd* **REASCODE=***rsn*

Explanation: RRSAF returned an unexpected return code when it was called to connect to DB2.

Severity: 12 (severe)

System action: Processing terminates.

User response: Refer to *DB2 Application Programming and SQL Guide* for information on the RRSAF RC, the return code, and the reason code.

DSNA622I *csect-name* **THE** *func-name* **FUNCTION CALL IS UNSUCCESSFUL,** *reason*

Explanation: A z/OS C/C++ function *func-name* returned an unexpected error code. *reason* is the run-time error message string that maps to the error code *errno*, followed by the *errno2* value.

Severity: 12 (severe)

System action: Processing terminates.

User response: Refer to C/C++ Run-Time Library Reference manual for information on the z/OS C/C++ function error codes.

DSNA623I *csect-name* **THE PROCEDURE** *procedure-name* **DID NOT COMPLETE SUCCESSFULLY, RETURN CODE=***xx*

Explanation: The procedure *procedure-name* did not complete successfully when it was internally invoked, returning the indicated return code value. This message is followed by the error message returned by the named procedure, if applicable.

Severity: 12 (severe)

System action: Processing terminates.

User response: Review the error message returned by the named stored procedure. Refer to the *DB2 Codes* for more information about the error message returned.

DSNA624I *csect-name* **THE STORED PROCEDURE ABENDED**

Explanation: The stored procedure abended and did not complete successfully.

Severity: 12 (severe)

System action: Processing terminates.

User response: Report the problem to the IBM Support Center.

DSNA625I *csect-name* **THE STORED PROCEDURE TASK WAS TERMINATED BY THE OPERATOR**

Explanation: The stored procedure did not complete successfully, because it was terminated by the operator.

Severity: 12 (severe)

System action: Processing terminates.

User response: Issue the SQL CALL statement again.

DSNA626I *csect-name* **THE** *func-name* **FUNCTION CALL IS UNSUCCESSFUL,** *parm-name*=*parm-value*, *reason*

Explanation: A z/OS C/C++ function *func-name* returned an unexpected error code when the parameter *parm-name* was passed to the function with the value *parm-value*. *reason* is the run-time error message string that maps to the error code *errno*, followed by the *errno2* value.

Severity: 12 (severe)

System action: Processing terminates.

User response: Refer to C/C++ Run-Time Library Reference manual for information on the z/OS C/C++ function error codes.

DSNA629I *csect-name* **THE PARAMETER** *parm-name* **RETURNED BY THE STORED PROCEDURE** *proc-name* **IS NOT VALID. INVALID REASON CODE=***xx*

Explanation: An invalid value was returned by the indicated procedure for the specified parameter. The cause of the invalid error is identified by the invalid reason code value in the message text. The specified parameter is not valid for the indicated reason:

3 value is null

Severity: 12 (severe)

System action: Processing terminates.

User response: Report the problem to the IBM Support Center.

DSNA634I *csect-name* **THE JES-MANAGED SYSOUT DATA SET IS EMPTY, JOBID=***jobid*

Explanation: A request was made to process SYSOUT data sets residing on JES spool for the indicated job identifier specified. However, the indicated job does not have any SYSOUT data sets.

Severity: 12 (severe)

System action: No action is taken with the specified job identifier.

User response: If the job action was requested by a stored procedure, correct the invalid job identifier on the SQL Call statement.

DSNA635I *csect-name* **AN INTERNAL ERROR OCCURRED.** *additional-info*

Explanation: An error, as indicated in *additional-info*, occurred.

Severity: 12 (severe)

System action: Processing terminates.

User response: Report the problem to the IBM Support Center.

DSNA636I *csect-name* **DB2 THREAD WAS CANCELED**

Explanation: The thread running the stored procedure was cancelled.

Severity: 12 (severe)

System action: Processing terminates.

User response: If necessary, issue the SQL CALL statement again.

DSNA637I *csect-name* **I/O ERROR ON DATA SET**

Explanation: An I/O error occurred while processing a data set or library member.

Severity: 12 (severe)

System action: Processing terminates.

User response: Look at the messages in the console log for more information. If necessary, report the problem to the IBM Support Center.

DSNA638I *csect-name* **THE SEQUENCE NUMBER** *seq* **IN ROW** *rownum* **OF TABLE** *table-name* **IS OUT OF SEQUENCE**

Explanation: The sequence number of the rows in the indicated table is not in ascending sequence, or the row sequence number does not start with 1.

Severity: 12 (severe)

System action: Processing terminates.

User response: Correct the sequence number of the rows in the indicated table.

DSNA639I *csect-name* **THE NUMBER OF ROWS** *table-size* **IN THE INPUT TABLE** *table-name* **AND THE NUMBER OF OBJECTS** *objnum* **SPECIFIED IN THE PARAMETER** *parm-name* **DO NOT MATCH**

Explanation: The number of rows in the indicated table did not match the number of objects specified in the named parameter.

Severity: 12 (severe)

System action: Processing terminates.

User response: Correct the number of objects specified in the indicated parameter, or repopulate the named table with the correct number of rows.

DSNA640I *csect-name* **SHUTDOWN DURATION** *xxxx* **REACHED, SHUTDOWN INITIATED**

Explanation: When the overall execution time reached the indicated shutdown duration, shutdown was initiated. When shutdown is initiated, current utility executions run to completion but no new utility will be started.

Severity: 4 (warning)

System action: Current utility executions are allowed to run to completion, but no new utility will be started.

User response: Determine which objects have not been processed due to the shutdown. Call SYSPROC.ADMIN_UTL_SCHEDULE again to process the unprocessed objects. If necessary, increase the SHUTDOWN_DURATION input parameter on the SQL CALL statement.

DSNA643I *csect-name* **INVALID CREDENTIALS, ERRNO=***errno*

Explanation: The stored procedure is unable to log in because the USERID or PASSWORD parameter in the SQL CALL statement is invalid.

The *csect-name* in the message identifies the CSECT that issued the message.

Severity: 12 (severe)

System action: The stored procedure terminates.

Programmer response: Specify valid credentials in the SQL CALL statement. If the problem persists, ask your security administrator to check the security configuration for the specified user.

DSNA645I *csect-name* **THE PARAMETER MAX_INVOCATIONS IS NOT 1 ALTHOUGH NO SCHEDULE IS DEFINED**

Explanation: This input parameter must be set to 1 when parameters INTERVAL, POINT_IN_TIME and TRIGGER_TASK_ID are null.

The *csect-name* in the message identifies the CSECT that issued the message.

Severity: 12 (severe)

System action: The stored procedure terminates.

Programmer response: Specify 1 for MAX_INVOCATIONS in the SQL CALL statement.

DSNA646I *csect-name* **THE PARAMETER** *parm-name* **HAS AN INVALID FORMAT, POS =** *xx*

Explanation: The specified parameter in the message has an invalid format. The error was detected at position *xx* in the parameter value.

The *csect-name* in the message identifies the CSECT that issued the message.

Severity: 12 (severe)

System action: The stored procedure terminates.

Programmer response: Correct the value for the specified input parameter in the SQL CALL statement.

DSNA650I *csect-name* **CANNOT CONNECT TO ADMIN SCHEDULER** *proc-name*

Explanation: The stored procedure or UDF cannot connect to the specified administrative scheduler. Either the name of the administrative scheduler is incorrect or the administrative scheduler is currently down.

The *csect-name* in the message identifies the CSECT that issues the message.

Severity: 12 (severe)

System action: The stored procedure terminates.

System programmer response: Verify the name of the administrative scheduler and change the DSN6SPRM ADMTPROC ZPARM if necessary. Verify that the specified administrative scheduler is up and running before calling this stored procedure again.

DSNA651I *csect-name* **DB2** *db2-ssid* **IS NOT ALLOWED TO CONNECT TO ADMIN SCHEDULER** *proc-name*

Explanation: The specified administrative scheduler does not accept requests from the specified DB2 subsystem.

The *csect-name* in the message identifies the CSECT that issues the message.

Severity: 12 (severe)

System action: The stored procedure terminates.

System programmer response: Verify the name of the
administrative scheduler and change the DSN6SPRM
ADMTPROC ZPARM if necessary. Verify that the
specified administrative scheduler is set up to accept
requests from the specified DB2 subsystem.

DSNA652I *csect-name* **THE USER** *user-name* **IS NOT**
ALLOWED TO ACCESS TASK *task-id*

Explanation: The specified user, which is the value to
which the special register CURRENT SQLID is set, is
not allowed to update or delete the specified task. This
user did not create the task and does not have
SYSOPR, SYSCTRL, or SYSADM authority.

The *csect-name* in the message identifies the CSECT that
issues the message.

Severity: 12 (severe)

System action: The stored procedure terminates.

Programmer response: Ensure that the DB2 special
register (CURRENT SQLID) is set to the same user that
added this task, or to a user that has SYSOPR,
SYSCTRL, or SYSADM authority, before calling this
stored procedure again.

DSNA653I *csect-name* **CANNOT FIND ZPARM**
parm-value

Explanation: The specified ZPARM is not defined.

The *csect-name* in the message identifies the CSECT that
issues the message.

Severity: 12 (severe)

System action: The stored procedure terminates.

System programmer response: Define the specified
ZPARM value.

DSNA654I *csect-name* **THE TASK** *task-id* **DOES NOT**
EXIST

Explanation: The specified task cannot be updated or
removed because it does not exist in the administrative
scheduler task list.

The *csect-name* in the message identifies the CSECT that
issues the message.

Severity: 12 (severe)

System action: The stored procedure terminates.

Programmer response: Specify an existing task to be
updated or removed.

DSNA655I *csect-name* **THE TASK** *task-id* **IS**
CURRENTLY EXECUTING

Explanation: The specified task cannot be updated or
removed because it is currently executing.

The *csect-name* in the message identifies the CSECT that
issues the message.

Severity: 12 (severe)

System action: The stored procedure terminates.

Programmer response: Wait for the completion of this
task before trying to update or remove it. Check the
execution status of the task using the UDF function
SYSFUN.ADMIN_TASK_STATUS.

DSNA656I *csect-name* **THE ADMIN SCHEDULER**
proc-name **TASK LIST IS FULL**

Explanation: The task list for the specified
administrative scheduler has reached its limit. No task
can be added until at least one task is removed.

The *csect-name* in the message identifies the CSECT that
issues the message.

Severity: 12 (severe)

System action: The stored procedure terminates.

Programmer response: Explicitly delete some tasks
that have already been executed and whose status is
not required anymore.

DSNA657I *csect-name* **THE ADMIN SCHEDULER**
proc-name **CANNOT ACCESS ITS TASK**
LISTS

Explanation: The specified administrative scheduler
cannot access any of its task lists to make the requested
changes. Tasks cannot be added, removed, or updated.

The *csect-name* in the message identifies the CSECT that
issues the message.

Severity: 12 (severe)

System action: The stored procedure terminates.

System programmer response: Check the operator
console messages for more details on this failure and
handle accordingly.

DSNA659I *csect-name* **THE TASK** *task-id* **TRIGGERS**
DEPENDENT TASKS

Explanation: The execution of the specified task
triggers the execution of one or more tasks defined in
the administrative scheduler task list. This task cannot
be removed.

The *csect-name* in the message identifies the CSECT that
issues the message.

Severity: 12 (severe)

System action: The stored procedure terminates.

Programmer response: The dependent tasks must be removed before trying to remove the specified task.

DSNA660I *csect-name* **AN INTERNAL ERROR HAS OCCURRED**

Explanation: The administrative scheduler could not access internal resource that should be available.

The *csect-name* in the message identifies the CSECT that issues the message.

Severity: 12 (severe)

System action: The stored procedure terminates.

System programmer response: Contact your IBM representative.

DSNA670I *csect-name* **THE ADMIN SCHEDULER** *proc-name* **IS STOPPING**

Explanation: The specified administrative scheduler is stopping.

The *csect-name* in the message identifies the CSECT that issues the message.

Severity: 0 (informational)

System action: The administrative scheduler is stopping.

DSNA671I *csect-name* **THE ADMIN SCHEDULER** *proc-name* **IS STARTING**

Explanation: The specified administrative scheduler is starting.

The *csect-name* in the message identifies the CSECT that issues the message.

Severity: 0 (informational)

System action: The administrative scheduler is starting.

DSNA672I *csect-name action-description proc-name* **NORMAL COMPLETION**

Explanation: The described action completed successfully for the specified administrative scheduler.

The *csect-name* in the message identifies the CSECT that issues the message.

Severity: 0 (informational)

System action: The administrative scheduler executed the specified action.

DSNA673I *csect-name action-description proc-name* **ABNORMAL TERMINATION**

Explanation: The described action was aborted because of an error in the specified administrative scheduler.

The *csect-name* in the message identifies the CSECT that issues the message.

Severity: 12 (severe)

System action: The administrative scheduler terminates.

Programmer response: Investigate other error messages that were displayed on the operator's console before this message.

DSNA674I *csect-name* **THE ADMIN SCHEDULER** *proc-name* **IS ALREADY ACTIVE**

Explanation: The administrative scheduler that is starting up has detected another address space with the same name that is already up and running on the system. Only one instance of the administrative scheduler is allowed to run.

The *csect-name* in the message identifies the CSECT that issues the message.

Severity: 12 (severe)

System action: The administrative scheduler that is starting up terminates.

DSNA675I *csect-name module-name* **COULD NOT RUN BECAUSE IT WAS NOT INVOKED FROM AN APF-AUTHORIZED LIBRARY**

Explanation: The specified program that starts the administrative scheduler could not run for one of the following reasons:

- *module-name* was not invoked from an APF-authorized library.
- One or more libraries in the STEPLIB or JOBLIB concatenation in the job that invokes *module-name* is not APF-authorized.

The *csect-name* in the message identifies the CSECT that issues the message.

Severity: 12 (severe)

System action: The program terminates.

User response: Notify the system programmer.

System programmer response: Ensure that *module-name* is link-edited with option AC(1) and resides in an APF-authorized library. Ensure that all libraries in the STEPLIB and JOBLIB concatenation for the job that invokes *module-name* are APF-authorized.

DSNA676I *csect-name* **SEVERE ERROR ENCOUNTERED WHILE STARTING OR STOPPING ADMIN SCHEDULER** *proc-name*

Explanation: A severe error has been detected in the specified administrative scheduler.

The *csect-name* in the message identifies the CSECT that issues the message.

Severity: 12 (severe)

System action: The administrative scheduler terminates.

Programmer response: Contact your IBM representative.

DSNA677I *csect-name* **THE ADMIN SCHEDULER** *proc-name* **CAN ONLY CREATE** *num-threads* **OF** *max-threads* **THREADS**

Explanation: There are not enough resources available to create all the requested threads *max-threads*. The specified administrative scheduler is only able to create a subset of *num-threads*.

The *csect-name* in the message identifies the CSECT that issues the message.

Severity: 12 (severe)

System action: The administrative scheduler terminates.

System programmer response: Increase system resources to allow the administrative scheduler to create the maximum number of threads needed.

DSNA678I *csect-name* **THE NUMBER OF TASKS TO BE CONCURRENTLY EXECUTED BY THE ADMIN SCHEDULER** *proc-name* **EXCEEDS** *max-threads*

Explanation: The current number of tasks to be executed concurrently exceeds the maximum number *max-threads* allowed by the specified scheduler. Remaining tasks will not be executed.

The *csect-name* in the message identifies the CSECT that issues the message.

Severity: 12 (severe)

System action: The administrative scheduler continues running, but some scheduled tasks cannot be executed.

Programmer response: Change the execution schedule of some tasks if possible.

DSNA679I *csect-name* **THE ADMIN SCHEDULER** *proc-name* **CANNOT ACCESS TASK LIST** *ds-name*, **REASON =** *reason*

Explanation: The specified administrative scheduler cannot access the specified VSAM data set *ds-name* containing the task list. The REASON value in the message text identifies this reason.

The *csect-name* in the message identifies the CSECT that issues the message.

Severity: 12 (severe)

System action: The administrative scheduler continues running with access to DB task list table only.

System programmer response: Verify the name of the VSAM data set and modify the admin scheduler started task procedure if necessary. Stop and restart the administrative scheduler with this new configuration.

According to the reason specified, make sure the following conditions are met.

- The VSAM data set exists.
- The VSAM data is allocated.
- The VSAM data can be accessed from this system.
- The user who started the admin scheduler is allowed to access it.

DSNA680I *csect-name* **THE ADMIN SCHEDULER** *proc-name* **TASK LIST** *list-name* **IS CORRUPTED**

Explanation: The specified administrative scheduler cannot recognize the internal structure of the task list in *list-name*.

The *csect-name* in the message identifies the CSECT that issues the message.

Severity: 0 (informational)

System action: The administrative scheduler tries recovering from the redundant task list.

Programmer response: Check later messages to see the result of the recovery process.

DSNA681I *csect-name* **THE ADMIN SCHEDULER** *proc-name* **TASK LIST** *list-name1* **AND** *list-name2* **ARE INCONSISTENT**

Explanation: The specified administrative scheduler detected that the task lists in the VSAM data set *list-name1* and in the DB2 table *list-name2* do not match.

The *csect-name* in the message identifies the CSECT that issues the message.

Severity: 0 (informational)

System action: The administrative scheduler tries recovering the older task list from the newer task list.

Programmer response: Check later messages to see

the result of the recovery process.

DSNA682I *csect-name* **THE ADMIN SCHEDULER**
proc-name **TASK LIST** *list-name* **IS**
SUCCESSFULLY RECOVERED

Explanation: The specified administrative scheduler
successfully recovered the corrupted or outdated task
list in *list-name* from the redundant task list.

The *csect-name* in the message identifies the CSECT that
issues the message.

Severity: 0 (informational)

System action: Both task lists are in a consistent state.

DSNA683I *csect-name* **THE ADMIN SCHEDULER**
proc-name **TASK LIST** *list-name*
CANNOT BE RECOVERED

Explanation: The specified administrative scheduler
could not recover the corrupted or outdated task list in
list-name from the redundant task list.

The *csect-name* in the message identifies the CSECT that
issues the message.

Severity: 12 (severe)

System action: The administrative scheduler continues
running with access to only one task list.

Programmer response: Contact your IBM
representative.

DSNA684I *csect-name* **DFLTUID CANNOT BE THE**
USER *user-name* **THAT STARTS THE**
ADMIN SCHEDULER *proc-name*

Explanation: The specified administrative scheduler
started task parameter (DFLTUID) is set to the specified
user name. This user is the same user that is associated
with the started task. For security reasons, this is not
allowed.

The *csect-name* in the message identifies the CSECT that
issues the message.

Severity: 12 (severe)

System action: The administrative scheduler does not
start.

System programmer response: Specify another user
name in the DFLTUID parameter. It is recommended
that the default execution user has minimal access
rights on the system and no access to the admin
scheduler resources.

DSNA690I *csect-name* **THE TASK** *task-id* **WAS**
RUNNING WHEN THE ADMIN
SCHEDULER *proc-name* **STOPPED**

Explanation: The specified task was executing when
the specified administrative scheduler was forced to
stop. The status of this task is set to complete even if
its real status is not known.

The *csect-name* in the message identifies the CSECT that
issues the message.

Severity: 12 (severe)

System action: The administrative scheduler recovers
from a blocked state.

System programmer response: Avoid forcing the
administrative scheduler to stop when tasks are
currently running. Make sure that no task is currently
running when trying to stop the administrative
scheduler.

DSNA691I *csect-name* **THE ADMIN SCHEDULER**
proc-name **CANNOT GENERATE A**
PASSTICKET FOR TASK *task-id*, **USER**
user-name

Explanation: The specified administrative scheduler
cannot generate a PassTicket to switch the security
environment in order to execute the specified task
under the authority of the specified user.

The *csect-name* in the message identifies the CSECT that
issues the message.

Severity: 12 (severe)

System action: The administrative scheduler does not
execute the specified task.

System programmer response: Check the security
configuration and make sure that the administrative
scheduler is allowed to create PassTickets for any user.

DSNA692I *csect-name* **THE ADMIN SCHEDULER**
proc-name **CANNOT LOG IN AS USER**
user-name **TO EXECUTE TASK** *task-id*

Explanation: The specified administrative scheduler
cannot switch the security environment in order to
execute the specified task under the authority of the
specified user.

The *csect-name* in the message identifies the CSECT that
issues the message.

Severity: 12 (severe)

System action: The administrative scheduler does not
execute the specified task.

System programmer response: Check the security
configuration and make sure that the specified user still
exists and can log in.

DSNA693I *csect-name* **THE ADMIN SCHEDULER**
proc-name **CANNOT CONNECT TO DB2**
db2-ssid **TO EXECUTE TASK** *task-id*

Explanation: The specified administrative scheduler
was unable to connect to the specified DB2 subsystem

when executing the specified stored procedure task.

The *csect-name* in the message identifies the CSECT that issues the message.

Severity: 12 (severe)

System action: The administrative scheduler does not execute the specified task.

Programmer response: Make sure that DB2 is up and running when a task is scheduled to run.

DSNA694I *csect-name* **THE ADMIN SCHEDULER** *proc-name* **CANNOT PROCESS PARAMETER PROCEDURE_INPUT OF TASK** *task-id*

Explanation: The parameter PROCEDURE_INPUT for the specified task does not specify an SQL SELECT statement that returns one row of data.

The *csect-name* in the message identifies the CSECT that issues the message.

Severity: 12 (severe)

System action: The administrative scheduler does not execute the specified task.

Programmer response: Correct the PROCEDURE_INPUT parameter for the specified task by invoking the ADMIN_TASK_SCHEDULE stored procedure with a valid PROCEDURE_INPUT parameter value.

DSNA8xx

Explanation: Messages in the DSNA800 series are intended for diagnosis and are documented in Part 8 of *DB2 Diagnosis Guide and Reference*.

Chapter 3. DSNB... messages

DSNB200I UPDATE VVDS FAILED. DSNAME = *dsn*, MMRC=*cccccccc* CONNECTION-ID=*xxxxxxxx*, CORRELATION-ID=*yyyyyyyyyyyy*, LUW-ID=*logical-unit-of-work-id* =*token*

Explanation: DB2 was unable to update the data set high-used RBA in the VSAM volume data set (VVDS) when it was ready to close a data set that was used by a RECOVER, LOAD, or REORG utility job. The error was detected by the media manager catalog update function that issues system message IEC161I. The *dsn* indicates the DSNAME of the data set. MMRC is the return code that DB2 received from the media manager catalog update service. Refer to *z/OS DFSMSdfp Diagnosis Reference* for a description of these codes.

The connection ID, correlation ID, and logical-unit-of-work ID (LUW-ID) identify the user or application that encountered the problem.

An LUW-ID of '*' indicates that the thread does not have an LUW-ID. If the LUW-ID is not an '*', the value is a logical-unit-of-work-id that is assigned to the thread followed by an '=' sign and a token value which can be used in place of logical-unit-of-work-id in any DB2 command that accepts logical-unit-of-work-id as input.

System action: DB2 continues closing this data set. However, the utility job is informed with a 00C2010C 'resource unavailable' reason code. The utility job is left in a failed state.

Operator response: Notify the system programmer.

System programmer response: To determine the cause of the error, refer to system message IEC161I in the appropriate MVS publication. Correct the error, and then notify the user to reinvoke the application.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 10, 11.

DSNB204I OPEN OF DATA SET FAILED. DSNAME = *dsn*

Explanation: A required data set of a DB2 table or index space could not be opened. The error was detected by the media manager CONNECT function that issues system message IEC161I. The *dsn* indicates the DSNAME of the data set.

After being issued for a given data set, this message is disabled for that data set.

System action: Buffer manager functions that detect this condition terminate.

Operator response: Notify the system programmer.

System programmer response: To determine the cause of the error, refer to system message IEC161I in the appropriate MVS publication. Correct the error, and reinvoke the application.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 5, 10, 12.

DSNB207I DYNAMIC ALLOCATION OF DATA SET FAILED. REASON = *rrrr* DSNAME = *dsn*

Explanation: A required data set of a DB2 table or index space could not be allocated. Reason code *rrrr*, which is returned by dynamic allocation, indicates the reason for the failure. Refer to the appropriate version of the *MVS Authorized Assembler Services Guide* (SA22-7608) publication for a description of this reason code. The *dsn* indicates the data set's DSNAME.

After being issued for a given data set, this message is disabled for that data set.

System action: Buffer manager functions that detect this condition terminate and pass a unique error return code.

Operator response: Notify the system programmer.

System programmer response: Correct the situation, and then reinvoke the application.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 5, 10, 12.

DSNB209I *csect-name*–BUFFER MANAGER TABLESPACE/INDEXSPACE CLEAN–UP ROUTINE (DSNB1CFC) HAS BEEN SCHEDULED FOR DBNAME = *dbn*, SPACENAME = *spn*, PSID = X'*psid*'

Explanation: DB2 detected an abnormal condition for one of the following reasons:

- The associated table space or index space contains modified pages that cannot be written to DASD for a period of time that is longer than expected. This can be caused by either an unusually long period of contiguous update or by an internal logic error.
- This message follows an abnormal termination due to inconsistent data or an incomplete update to the page.

DB2 scheduled the table space or index space cleanup routine (DSNB1CFC) for the associated table space or

© Copyright IBM Corp. 1983, 2008

index space. DSNB1CFC acquires a drain lock to ensure that no updates are in progress. Data is inaccessible to users during the cleanup process.

For recoverable inconsistent pages with unbalanced write intents, DB2 initiates the recovery process and reconstructs the pages using the DB2 log. For other modified pages with unbalanced write intents, the buffer manager forces the pages to be written back to DASD and then makes the buffer space available for reassignment.

The *dbn* indicates the database name with which the table space or index space is associated. The *spn* is the name of the table space or index space that must be cleaned up. The *psid* is the internal identifier for the table space or index space; it is used to associate this message with follow-on message DSNB221I, which might be issued.

System action: DB2 is ready to clean up the associated table space or index space. However, before any cleanup is initiated, DB2 requests a drain on the table space or index space. DB2 waits indefinitely for this lock. Therefore, all activities against the table space and/or index space must be quiesced before the cleanup process can proceed. If the associated object is an index space, all activities against the table space for which the index space is defined must also be quiesced. After the cleanup process completes, the table space or index space is automatically made accessible to users.

User response: If the clean up activity does not appear to be completed, DB2 might still be waiting for a drain lock on the table space. Issue a -DISPLAY DATABASE CLAIMERS command for the table space in question (for index spaces, this would be the associated table space) to determine whether the clean up routine is still waiting for a lock and if so, to determine the holders of the table space lock. If the jobs are holding the table space lock for a very long duration (for instance, wait-for-input), it might be necessary to cancel those jobs in order to allow the clean up process to proceed.

DSNB212I *csect-name*–**RECOVER PAGE STARTING DBNAME =** *dbn* **, SPACENAME =** *spn* **, PAGE NUMBER = X'***pno***'**

Explanation: The buffer manager (BM) subcomponent of DB2 discovered an inconsistent page with unbalanced write intents. DB2 is ready to do online recovery on the page using the DB2 log. The *dbn* indicates the associated database name. The *spn* indicates the table space name or index space name. The *pno* is the page number (in hexadecimal) within the table space or index space that is logically inconsistent.

System action: Processing continues.

DSNB215I *csect-name*–**ONLINE RECOVERY FOR AN INCONSISTENT PAGE HAS BEEN COMPLETED FOR DBNAME =** *dbn* **, SPACENAME =** *spn* **, PAGE NUMBER = X'***pno***'**

Explanation: The recovery process for an inconsistent page completed. The *dbn* indicates the associated database name. The *spn* indicates the table space name or index space name. The *pno* is the page number (in hexadecimal) within the table space or index space that was logically inconsistent.

System action: Processing continues.

DSNB217I *csect-name*–**ONLINE RECOVERY FOR AN INCONSISTENT PAGE WAS UNSUCCESSFUL FOR DBNAME =** *dbn* **, SPACENAME =** *spn* **, PAGE NUMBER = X'***pno***'**

Explanation: The recovery process for an inconsistent page failed. The page is written to DASD as logically inconsistent. This message is issued in response to message DSNB212I, which was issued to indicate the start of the recovery process. The *dbn* indicates the associated database name. The *spn* indicates the table space name or index space name. The *pno* is the page number (in hexadecimal) within the table space or index space that was logically inconsistent.

System action: The page is marked as logically inconsistent and is unavailable until it is repaired. The execution unit driving this function proceeds normally.

Problem determination: A DSNI012I message might be issued before this message to indicate the same failure. If it is, a 00C90102 abend dump might help determine the cause of this failure. Otherwise, see SYS1.LOGREC for any abend that occurred between the associated DSNB212I message and this message. Use the time stamp on the log records to find this abend.

DSNB219I *csect-name*–**BUFFER MANAGER TABLESPACE/INDEXSPACE CLEAN–UP PROCESS HAS BEEN COMPLETED FOR DBNAME =** *dbn* **, SPACENAME =** *spn*

Explanation: This message indicates that the buffer manager (BM) table space or index space cleanup process completed for the associated table space or index space. Refer to message DSNB209I for the reasons why the buffer manager schedules the table space or index space cleanup process. The *dbn* indicates the associated database name. The *spn* indicates the table space name or index space name.

System action: Processing continues.

DSNB221I *csect-name*–**BUFFER MANAGER TABLESPACE/INDEXSPACE CLEAN–UP PROCESS WAS NOT REQUIRED FOR PSID = X'***psid***'**

Explanation: This message indicates that the buffer manager (BM) table space or index space cleanup process was bypassed for the associated table space or index space. After quiescing all other activities against the table space or index space, the buffer manager discovered that the table space or index space is no longer in active mode; that is, it was closed.

The abnormal condition detected earlier was resolved by buffer manager close page set processing. Refer to the corresponding message, DSNB209I, using *psid* as the matching field, for the reasons the buffer manager schedules the table space or index space cleanup process. The *psid* is the internal identifier for the associated table space or index space.

System action: Processing continues.

DSNB223I *csect-name*–**BUFFER MANAGER TABLESPACE/INDEXSPACE CLEANUP PROCESS WAS SKIPPED FOR PSID = X'***psid***'**

Explanation: This message indicates that the table space or index space cleanup process was bypassed for the associated table space or index space. During the process of quiescing all other activities against the table space or index space, an error that is not recoverable was detected. The error might be caused by any of the following conditions:

- An out-of-storage condition was detected by IRLM.
- The table space or index space was dropped because of a backout.

It is assumed that the abnormal condition detected earlier can later be resolved by the close page set processing. Refer to the corresponding message DSNB209I (use *psid* as the matching field) for the reasons for scheduling the table space or index space cleanup process. The *psid* specifies the internal identifier for the associated table space or index space.

System action: Processing continues.

System programmer response: The associated table space or index space might contain inconsistent data. Issue the DB2 STOP TABLESPACE or INDEXSPACE command to allow close page set processing to perform cleanup for the associated table space or index space.

DSNB224I *csect-name*– **BUFFER MANAGER I/O ERROR DURING** *function*, **DBNAME=***dbn*,**SPACENAME=***snm*, **DATA SET NUMBER=***dsno*, **PAGE NUMBER=X'***pno***', MM ERROR FLAG=X'***aa***', CSW STATUS=X'***bbbb***', SENSE BYTES=X'***cccc***', SEEK ADDRESS=X'***dddddddddddddddd***'**

Explanation: An I/O error was encountered by the buffer manager (BM) subcomponent of DB2. The function specified in the message is READ, WRITE, or FORMAT. The message gives the database name, table space name or index space name, data set number, and page number (in hexadecimal) of the page for which the I/O operation was being performed. The remaining information is error information returned by the media manager and is used for error analysis by IBM support personnel.

System action: The system action depends on the function being performed. If the function is WRITE, or if the function is READ and the operation was performed during restart or rollback, the I/O error page range of the data set is updated and message DSNU086I is issued. In most cases (READ, synchronous WRITE, FORMAT), a 'resource not available' is returned to the user with a reason code that identifies the nature of the problem.

After being issued 10 times for a given data set, this message is disabled for that data set until it is closed and reopened.

Operator response: Notify the system programmer.

System programmer response: See documentation in *DB2 Codes* for the reason code returned to the user in the SQLCA or in utility messages. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: See the Problem Determination section of the reason code returned to user.

Check the MVS console log for an IOS error message indicating the type of I/O error that occurred. If an IOS message was issued, follow the problem determination directions for that message.

Collect the following diagnostic items listed in Appendix A, “Problem determination,” on page 659: 1, 5.

DSNB225I *csect-name* **- BUFFER MANAGER I/O ERROR DURING** *function* **DBNAME=***dbn*,**SPACENAME=***snm*, **DATA SET NUMBER=***dsno*, **MM ERROR CODES=X’***aaaaaaaa***’ DB2 REASON CODE=X’***bbbbbbbb***’**

Explanation: DB2 encountered an I/O error. The function specified in the message is READ, WRITE, CASTOUT, or FORMAT. The message contains the database name, table space name or index space name, data set number, media manager error information returned at the time of the failure, and the DB2 reason code associated with this failure.

System action: Except for a castout error, the

application terminates with abend X'04E' and reason code 00C200A4 or 00C200C0. If an error occurs during castout, processing continues.

Operator response: Notify the system programmer and follow the directions given for the reason code described in *DB2 Codes*.

System programmer response: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: This message is issued before the occurrence of abend X'04E' and reason code 00C200A4 or 00C200C0 to provide information related to the media manager failure.

Check the MVS console log for an IOS error message indicating the type of I/O error that occurred. If an IOS message was issued, follow the problem determination directions for that message.

Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 5, 10, 12.

DSNB226I *csect-name* **- BUFFER MANAGER PAGE GET ROUTINE DETECTED AN INVALID PAGE IN DBNAME=***dbn* **SPACENAME=***snm* **DATA SET NUMBER=** *dsno* **PAGE EXPECTED=X'***aaaaaa***' PAGE NUMBER FOUND=X'***bbbbbb***'**

Explanation: An unexpected page was received during buffer manager (BM) get page processing. The page set involved is identified by the database name, space name, and data set number. The page in error is identified by page expected *aaaaaa*. Page number *bbbbbb* was found in the buffer that should contain the expected page.

System action: The application is terminated with abend X'04E' and reason code 00C200F7.

Operator response: Notify the system programmer and follow the directions given for reason code 00C200F7 in *DB2 Codes*

System programmer response: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: This message is issued before the occurrence of abend X'04E' and reason code 00C200F7 to provide information about the page set involved.

Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 5, 10, 12, 64.

DSNB227I *csect-name* **- DFHSM RECALL FAILED FOR DSNAME=***dsn* **HSMRC=***aaaaaaaa* **HSMRSN=***bbbbbbbb*

Explanation: The attempt to recall a migrated data set failed. DSNAME identifies the data set. HSMRC and HSMRSN are the return code and reason code given by DFHSM.

System action: A 'resource not available' code is returned to the user.

Operator response: Notify the system programmer.

System programmer response: Use the information provided in the message to determine the reason for the DFHSM recall failure.

Problem determination: Refer to *DB2 Codes* for an explanation of the return and reason codes contained in this message. Refer to the DFHSM message ARC11nnI, where nn is the value of HSMRC.

DSNB228I *csect-name* **GROUP BUFFER POOL** *gbpname* **CANNOT BE ACCESSED FOR** *function* **MVS IXLCACHE REASON CODE=***reason*

Explanation: While accessing group buffer pool *gbpname* for function *function* DB2 received an error from the MVS IXLCACHE request.

function READ or WRITE

reason Reason code from the MVS IXLCACHE request

System action: The system action depends on the function being performed. If the function is a WRITE, or a READ performed during rollback or DB2 restart, the page is added to the logical page list (LPL) for the data set and message DSNB250E is issued. Otherwise, 'resource unavailable' is returned to the user with a reason code that identifies the specific problem.

System programmer response: See *MVS/ESA Programming: Sysplex Services Reference* for an explanation of the MVS IXLCACHE reason code.

DSNB229I *csect-name* **UNEXPECTED ERROR WHEN CHECKING FOR REQUEST COMPLETION FOR GROUP BUFFER POOL** *gbp-name* **MVS IXLFCOMP REASON CODE=***reason*

Explanation: DB2 received an unexpected error from MVS when checking for the completion of a write request to the secondary group buffer pool for duplexing. *gbp-name* indicates the name of the group buffer pool. If the group buffer pool is duplexed and the error was encountered on a request to the secondary group buffer pool, then *gbp-name* indicates the name of the group buffer pool suffixed with '-SEC'.

reason is the reason code from the MVS IXLFCOMP service.

System action: DB2 returns the group buffer pool back to simplex mode. The secondary group buffer pool is deallocated.

System programmer response: See *MVS/ESA Programming: Sysplex Services Reference* for an explanation of the MVS IXLFCOMP reason code.

DSNB232I *csect-name* **- UNEXPECTED DATA SET LEVEL ID ENCOUNTERED - LEVEL ID =***lvlid* **EXPECTED LEVEL ID=***xlvlid* **DSN=***dsn*

Explanation: A required data set of a DB2 table space could not be opened. DB2 encountered an unexpected data set level ID.

lvlid The unexpected level ID from the data set. If DB2 was unable to retrieve the level ID from the data set (for example, if the data set had been deleted and redefined), the level ID is shown as "X''".

xlvlid The level ID recorded by DB2 previously and is the expected level ID for opening the data set

dsn The DSNAME of the data set.

This problem could be the result of data sets that were restored to a level that is inconsistent with other data sets, inconsistent with the DB2 log, or both.

System action: DB2 sends back a 'resource unavailable' SQL code to the application and a reason code explaining the error.

Operator response: Notify the system programmer.

System programmer response: To enable access to a data set reported as down level, take one of the following actions:

- Restore the page set or partition to the proper level (use DSN1COPY, DFHSM, or another tool).

  Identify the level ID of a copy of the page set or partition by running the DSN1PRNT(FORMAT) utility. The level ID is found in the header page of the page set or partition.
- Recover the down level page set or partition to currency by running the RECOVER or RECOVER LOGONLY utility.
- Accept the down levelling by running the REPAIR utility with the LEVELID option.

  This option is only valid if DB2 is able to access the level ID (*lvlid* is not "X''").

  Inconsistent data problems might result from this action and are the user's responsibility.
- Replace the contents of the page set or partition using the LOAD REPLACE utility.
- Explicitly restore the page set to a prior point in time with the RECOVER utility.

If the expected level ID is less than the encountered level ID, it is possible that the object identified in the message was ported from another system improperly or it might indicate that DSNDB01.SYSLGRNX is down level. If the problem is in the object identified in the message, the actions listed above can be used to allow access to the data. If the problem is in SYSLGRNX, this is a serious situation that compromises the recoverability of all objects managed by DB2. Investigate whether SYSLGRNX might have inadvertently been restored to a point in time inconsistent with the rest of the DB2 system.

DSNB250E *csect-name* **A PAGE RANGE WAS ADDED TO THE LOGICAL PAGE LIST DATABASE NAME =** *dbn* **SPACE NAME =** *spn* **DATA SET NUMBER =** *dsno* **PAGE RANGE =** *lowpg* **TO** *highpg* **START LRSN =** *startlrsn* **END LRSN =** *endlrsn* **START RBA =** *startrba* **LPL TRACE ID =** *traceid* **LPL REASON TYPE =** *reasontype*

Explanation: DB2 added a page range to the logical page list (LPL) for the specified data set because of a problem with the pages in the page range. The *startlrsn* and *endlrsn* give the log range needed to recover the pages in the page range. These values are RBA values unless DB2 is in the data sharing environment, in which case these values are LRSN values. In the data sharing environment, *startrba* indicates the starting RBA value of the recovery log range if the value is not zero. The *startrba* only applies to log records for the DB2 member that issued the message. The zero value for *endlrsn* in this message refers to the current end of the recovery log. The *traceid* is for IBM serviceability. The *reasontype* explains why the pages were added into LPL; the following are the reason types:

DASD DB2 encountered a DASD I/O error when trying to read or write pages on DASD.

LOGAPPLY
DB2 cannot apply log records to the pages.

GBP DB2 cannot successfully read nor externalize the pages from or to the group buffer pool because of link or structure failure, the GBP is in rebuild status, or the GBP was disconnected.

LOCK DB2 cannot get the required page latch or page P-lock on the pages.

CASTOUT
The DB2 Castout processor cannot successfully cast-out the pages.

System action: The pages are added to the LPL, and DB2 initiates the Automatic LPL Recovery. The pages remain inaccessible until they are recovered with either

the START DATABASE command with the SPACNAM option, the RECOVER utility, the LOAD utility, or the Automatic LPL Recovery. Users attempting to access a page in the LPL receive SQLCODE -904 with reason code 00C20206. If MVS detects a problem during group buffer pool access processing, DB2 writes a SYS1.LOGREC entry describing the error.

System programmer response: Check the reason and follow the appropriate actions, based on the console message. To recover pages from the LPL, perform one of the following actions:

- Issue the -START DATABASE command with the SPACENAM option
- Run the RECOVER or LOAD utility

DSNB301E *csect-name* **GROUP BUFFER POOL** *gbpname* **CANNOT BE CONNECTED DB2 REASON CODE =** *reason1* **MVS IXLCONN REASON CODE =** *reason2*

Explanation: DB2 was not able to connect to the named group buffer pool. This message is generally issued once for each DB2 subsystem for each DB2 reason code, except 00C20204, until the failure condition is corrected.

reason1 DB2 reason code

reason2 MVS IXLCONN reason code

For DB2 reason code 00C20204, which indicates a problem during MVS IXLCONN processing, this message is issued once for each DB2 subsystem for each IXLCONN reason code until the failure condition is corrected. If the DB2 reason code is not 00C20204, then *reason2* contains zeros. See *MVS/ESA Programming: Sysplex Services Reference* for an explanation of MVS IXLxxxx reason codes

System action: If the connection was requested on behalf of an access request by an SQL user on this DB2, SQLCODE -904 (resource unavailable) is returned to the SQL user with the DB2 reason code shown in this message.

If the connection was requested on behalf of an SQL user on another DB2 in the data sharing group, SQLCODE -904 (resource unavailable) is returned to the SQL user with DB2 reason code 00C20252. This indicates that the requested table space or index space is unavailable because another DB2 cannot connect to the group buffer pool.

If the MVS IXLCONN reason code is xxxx0C08 (structure allocation failed because there was no suitable facility to allocate based on the preference list in the MVS CFRM administrative policy), or if the MVS IXLCONN reason code is xxxx084C (user does not have proper Security Access Facility (SAF) authorization), then IFCID 0250, if active, contains additional diagnostic information.

DSNB302I *csect-name* **GROUP BUFFER POOL** *gbpname* **IS ALLOCATED IN A VOLATILE STRUCTURE**

Explanation: The DB2 group buffer pool structure is volatile.

One DB2 member issues this message whenever the structure has the volatile attribute. DB2 detects this either on the first connection to the group buffer pool or when the attribute changes later. DB2 always requests nonvolatile group buffer pools but allows allocation in a volatile structure if that is all that is available.

System action: DB2 processing continues.

System programmer response: To cause the group buffer pools to be allocated in nonvolatile coupling facilities, ensure that:

- At least a subset of your coupling facilities are configured correctly for nonvolatility.
- The MVS CFRM policy correctly specifies the nonvolatile coupling facilities in the preference list for the group buffer pool coupling facility structures.

Even when these things are correctly done, there might be instances when a suitable coupling facility in the preference list cannot be found (for example, not enough free space), and therefore allocation occurs in a different coupling facility. MVS tries to allocate the structure in the facility that best meets the structure allocation request as follows:

1st choice:
Coupling facility that meets the volatility requirement, the space requirement, and the exclusion list requirement.

2nd choice:
Coupling facility that meets the volatility requirement and the storage requirement.

3rd choice:
Coupling facility that has the best fit (most available free space) even though this might be less than the amount requested.

4th choice:
Structure cannot be allocated.

IFCID 0250, if activated, contains a list of the coupling facilities on which allocation was attempted.

DSNB303E *csect-name* **A LOSS OF CONNECTIVITY WAS DETECTED TO GROUP BUFFER POOL** *gpbname*

Explanation: Connectivity to group buffer pool *gpbname* was lost.

System action: DB2 disconnects from the coupling facility structure with REASON=FAILURE.

The action taken in response to this problem depends on the attempted function, as follows:

- For a coupling facility read: SQLCODE -904 (resource unavailable) is returned to the SQL user with a reason code indicating the specific error.
- For a coupling facility read during rollback, backout, or restart: The inaccessible page is added to the logical page list (LPL) for the page set. Message DSNB250E is issued for the first page of each data set added to the LPL.
- For a coupling facility read during castout: The castout operation cannot complete.
- For a coupling facility write: The changed page is added to the LPL for the page set.

DSNB304I *csect-name* **GROUP BUFFER POOL** *gbpname* **WAS SET TO 'DAMAGE ASSESSMENT PENDING' STATUS**

Explanation: A coupling facility structure for a group buffer pool was damaged.

System action:

- DB2 puts the group buffer pool into 'damage assessment pending' (DAP) status. Message DSNB304I indicates the DAP status. The group buffer pool cannot be accessed until DAP status is cleared.
- The DB2 damage assessment process determines which table spaces and index spaces are using the failed group buffer pool and puts those spaces into 'group buffer pool recovery pending' (GRECP) status.
- After DB2 sets all the necessary spaces to GRECP status, the DAP status for the group buffer pool is cleared. Message DSNB305I indicates that DAP status is cleared.

DSNB305I *csect-name* **'DAMAGE ASSESSMENT PENDING' STATUS WAS CLEARED FOR GROUP BUFFER POOL** *gbpname*

Explanation: Damage assessment completed and the 'damage assessment pending' (DAP) status is cleared for the group buffer pool (GBP). At this point, the group buffer pool status is no longer restricted and DB2 attempts to reconnect to the group buffer pool on the next reference to a GBP-dependent page set within the associated buffer pool.

Message DSNB304I precedes this message.

DSNB306I *csect-name* **THE REQUESTED LENGTH OF THE LOCAL CACHE VECTOR COULD NOT BE ALLOCATED FOR GROUP BUFFER POOL** *gbpname* **REQUESTED LENGTH =** *length1* **ALLOCATED LENGTH =** *length2*

Explanation: DB2 connected to the group buffer pool indicated by *gbpname*, but the MVS Cross-system Extended Services (XES) component could not allocate the requested size of the local cache vector (LCV) in the hardware system area (HSA). DB2 requires an LCV size that is large enough to handle all pages in the member's buffer pool.

length1 indicates the LCV size (in number of bits) that DB2 requested. *length2* indicates the LCV size (in number of bits) that XES was able to allocate.

To prevent flooding the operator console, this message is only issued once per DB2 member per group buffer pool until the condition is corrected.

System action: DB2 disconnects from the GBP and issues message DSNB301E with reason code 00C20208, if it has not yet been issued.

SQLCODE -904 is returned to the user with a reason code of 00C20208.

System programmer response: There is not enough HSA storage on the processor to support the size of the local cache vector. The details of HSA allocation are machine-dependent and are defined by the hardware. To correct this problem, take one or both of the following actions:

- Consult the documentation for your processor hardware to determine if the HSA storage allocation can be increased, and if so, how to do it.
- Modify the software configuration in your parallel sysplex to reduce the number or size of the vectors that are in use by other coupling facility structure users. Consider the following possible solutions:
  - Use the DB2 ALTER BUFFERPOOL command to reduce the size of this local buffer pool. This means that you might have to reduce the size of the virtual pool.
  - Use the DB2 ALTER BUFFERPOOL command to reduce the size of another local buffer pool that is already connected to a group buffer pool. This means that you might have to reduce the size of the virtual pool. If you make a connected buffer pool smaller, DB2 can usually contract the size of the corresponding LCV dynamically, making HSA storage available. However, DB2 cannot always do the dynamic LCV contraction. In cases where DB2 cannot dynamically contract the LCV size, you can use the DB2 ALTER BUFFERPOOL command to either force a deallocation of the LCV or to force a contraction of the LCV as follows:
    1. Redefine the virtual pool to be smaller than the original size.
    2. Take one of the following actions:
       a. Make the virtual pool very small (perhaps 50 pages or less). Making the virtual pool sufficiently small should let DB2 dynamically contract the LCV, or
       b. Delete the virtual pool to let DB2 disconnect from the group buffer pool. This

frees the entire LCV associated with that member's connection to that group buffer pool.

– Consolidate your group buffer pool-dependent page sets into fewer buffer pools to reduce the number of group buffer pool connections.

– Consult the documentation for the other software components that might be using a coupling facility cache or list structure with an associated LCV to determine whether it is possible to either delete the connection or to reduce the size of the LCV associated with the connection.

DSNB307I *csect-name* **GROUP BUFFER POOL** *gbpname* **WAS SUCCESSFULLY ALLOCATED BUT THE EXCLUSION LIST WAS NOT HONORED**

Explanation: DB2 successfully connected to the group buffer pool and allocated the group buffer pool in a coupling facility. However, the group buffer pool was allocated in a coupling facility containing a structure in the exclusion list.

System programmer response: If you require that the exclusion list be honored, you must ensure that there is sufficient space in the coupling facilities that are not in the exclusion list. IFCID 0250 provides information about which facilities were attempted.

DSNB308I *csect-name* **GROUP BUFFER POOL** *gbpname* **WAS SUCCESSFULLY ALLOCATED BUT THE ALLOCATED SIZE OF** *alloc_size* **IS LESS THAN THE REQUESTED SIZE OF** *req_size*

Explanation: DB2 successfully connected to the group buffer pool and allocated the group buffer pool in a coupling facility. However, the allocated size of the group buffer pool was less than the requested size in the MVS CFRM administrative policy because there is not enough space in the preference list facilities. Both *alloc_size* and *req_size* are in 4KB blocks.

System programmer response: IFCID 0250 provides information about which coupling facilities were used for attempted allocation.

DSNB309I *csect-name* **GROUP BUFFER POOL** *gbpname* **HAS BEEN DISCONNECTED WITH A REASON OF '***reason***'**

Explanation: DB2 disconnected from the named group buffer pool.

reason indicates one of the following values:

NORMAL
DB2 disconnected the group buffer pool normally. MVS deletes the connection.

FAILURE
DB2 disconnected the group buffer pool abnormally. MVS puts the connection into a 'failed-persistent' state. DB2 disconnects the GBP with a reason of FAILURE when some type of error has been detected, and castout processing cannot proceed. Usually the error involves a connectivity failure to either the coupling facility or DASD. This message is preceded by message DSNB303E, and probably a series of DSNB311I and DSNB312I messages.

If this message is issued because connectivity was lost to the group buffer pool, this message signifies that this DB2 member has completed its processing of the error.

System programmer response: See the system programmer response for message DSNB311I or DSNB312I.

DSNB310I *csect-name* **GROUP BUFFER POOL** *gbpname* **IS ALLOCATED IN A NONVOLATILE STRUCTURE**

Explanation: The coupling facility storage in which the group buffer pool is allocated was previously volatile, but is now nonvolatile. For availability reasons, we recommend that DB2 group buffer pools be allocated in nonvolatile coupling facility storage. When DB2 connects a group buffer pool, it requests that the backing coupling facility structure be allocated in nonvolatile storage. If the request for nonvolatile storage is not honored, DB2 issues message DSNB302I.

System action: DB2 processing continues.

DSNB311I *csect-name* **DBNAME** *database* **SPACE NAME** *spacename* **HAS PAGES IN THE LOGICAL PAGE LIST**

Explanation: This message is issued after a loss of connectivity to a group buffer pool was detected. The message indicates that the named table space or index space has one or more pages in the logical page list (LPL) that need to be recovered. This message is issued after DB2 has completed its processing for the table space or index space; this is the earliest point at which you can recover LPL pages for that table space or index.

System action: The pages in the LPL are not accessible until they have been recovered.

System programmer response: Issue the START DATABASE command with the named table space or index to recover the LPL pages. The START DATABASE command must be issued from a DB2 member that can connect to the group buffer pool. Alternatively, the RECOVER utility or the LOAD utility can be used to recover the LPL pages.

DSNB312I *csect-name* **DBNAME** *database* **SPACE NAME** *spacename* **PARTITION** *part-number* **HAS PAGES IN THE LOGICAL PAGE LIST**

Explanation: This message is issued after a loss of connectivity to a group buffer pool was detected. The message indicates that a partition of the named table space or index has one or more pages in the logical page list (LPL) that need to be recovered. This message is issued after DB2 has completed its processing for the partition; this is the earliest point at which you can recover the LPL pages for that partition.

System action: The pages in the LPL are not accessible until they have been recovered.

System programmer response: Issue the START DATABASE command on the table space or index space partition to recover the LPL pages. The START DATABASE command must be issued from a DB2 member that can connect to the group buffer pool. Alternatively, the RECOVER utility or the LOAD utility can be used to recover the LPL pages.

DSNB313I *csect-name* **GROUP BUFFER POOL** *gpbname* **TO BE DISCONNECTED REASON=***reason* **LOSSCONN PERCENTAGE=***percentage*

Explanation: This DB2 member is disconnecting from group buffer pool *gpbname* because it has lost connectivity to the coupling facility in which the group buffer pool is allocated.

reason is one of the following:

NOSFMPOL
MVS cannot determine the scope of the loss of connectivity because there is no active SFM policy with CONNFAIL(YES).

LOSSCONN
MVS determined that the scope of the loss of connectivity was less than 100%.

NOXESEVL
MVS started to try to determine the scope of the loss of connectivity, but some condition occurred which prevented MVS from being able to complete the evaluation. One example of such a condition would be that the SFM policy became inactive while MVS was trying to do the evaluation.

If *reason* is LOSSCONN, then *percentage* indicates the percentage of lost connectivity that was calculated by MVS. MVS evaluates the scope of the loss of connectivity in terms of SFM weights associated with the systems which have lost connectivity and which have active connectors to the coupling facility structure (if there are systems that do not have an active connector to the coupling facility structure, then those systems are not factored into the "percentage loss of connectivity" calculation).

If *reason* is not LOSSCONN, then *percentage* is not applicable (it should appear as zero).

System action: The DB2 member disconnects the group buffer pool, putting its connection into a "failed-persistent" state.

If there are transactions that are in progress at the time of the connectivity failure, DB2 might need to add pages to the logical page list (LPL).

System programmer response: Use the START DATABASE command to recover the LPL pages. If DSNDB01 or DSNDB06 objects have LPL pages, you might need to recover these first, before proceeding with LPL recovery for other objects.

DSNB314I *csect-name* **DAMAGE ASSESSMENT TO BE TRIGGERED FOR GROUP BUFFER POOL** *gpbname* **REASON=***reason*

Explanation: DB2 is about to initiate the damage assessment function for group buffer pool *gpbname* because of a detected error in either the coupling facility in which the group buffer pool was allocated, or in the connectivity from the MVS systems to the coupling facility.

reason is one of the following:

100%LCON
An error has occurred in a coupling facility or in the attachments of MVS systems to a coupling facility which has caused all DB2 members to lose connectivity to the named group buffer pool.

STRFAIL
There was an error in the coupling facility in which the group buffer pool was allocated which caused a failure of the coupling facility structure.

CONERROR
The named group buffer pool is currently allocated in a coupling facility, but DB2 cannot connect to the group buffer pool because the system on which it is running does not currently have connectivity, and MVS has indicated that currently no other member is actively connected and has connectivity.

DAP The named group buffer pool is in damage assessment pending (DAP) state.

System action: DB2 triggers the damage assessment function to ensure that any changed pages that might have been cached in the affected group buffer pool get marked as "needing recovery" (the objects to which the pages belong are marked in GRECP state). This message should be immediately followed by a

DSNB304I message indicating that the group buffer pool has been entered into damage assessment pending (DAP) state.

If there are transactions that are in progress at the time of the connectivity failure, DB2 might need to add pages to the logical page list (LPL).

There is one case where DB2 triggers damage assessment for a group buffer pool, but no DSNB314I message is issued. This case is where the group buffer pool was manually deleted (via MVS SETXCF FORCE commands) while all connections to the group buffer pool were failed-persistent (e.g. all DB2 members were down). In this case, the next DB2 member to connect to the group buffer pool will cause MVS to try to reallocate a new instance of the group buffer pool in some coupling facility in the CFRM policy preference list. If the allocation succeeds, then DB2, by doing a coupling facility structure version id comparison, will detect that the previous instance of the group buffer pool had been lost, and will trigger damage assessment. But you will not receive the DSNB314I message in this case, nor will you receive the DSNB304I or DSNB305I messages indicating that the group buffer pool has been put into damage assessment pending (DAP) state. You will only receive a series of DSNB322I and/or DSNB323I messages indicating those pagesets or partitions that have been put into GRECP state by the damage assessment process.

System programmer response: Use the START DATABASE command to recover the LPL pages and GRECP objects. If DSNDB01 or DSNDB06 objects are in GRECP state or have LPL pages, you might need to recover these first, before proceeding with GRECP or LPL recovery for other objects.

DSNB315I *csect-name* **GROUP BUFFER POOL** *gpbname* **IS ALLOCATED AS GBPCACHE** *cacheatt*

Explanation:

YES The group buffer pool will be used for data caching.

NO The group buffer pool will not be used for data caching. It will be used for cross invalidation only.

One DB2 member issues this message whenever the structure is allocated. DB2 detects this on the first connection to the group buffer pool either in the mainline processing or during group buffer pool rebuild.

System action: DB2 processing continues.

DSNB319A *csect-name* **THERE IS A SHORTAGE OF SPACE IN GROUP BUFFER POOL** *gbp-name*

Explanation: The amount of storage in the group buffer pool (GBP) coupling facility structure available for writing new pages is low. The GBP is 75 percent full with changed data. As the GBP fills with changed data, the castout thresholds are continuously reached, and the castout engines must continuously work to attempt to keep the number of changed pages in the GBP below the castout thresholds. Also, the more changed data the GBP contains, the less efficiently it operates. Completely running out of GBP storage can result in transaction delays and in pages being added to the logical page list (LPL).

This message stays on the MVS console and is highlighted.

System action: DB2 processing continues.

This message might remain highlighted and might not be automatically deleted in a pre SP5.2.1-JES3 environment.

Operator response: Contact the system programmer to fix the storage problem.

System programmer response:

You can relieve the GBP storage shortage by taking one or more of the following actions:

- Use the ALTER GROUPBUFFERPOOL command to lower the CLASST and GBPOOLT castout thresholds.
- Use the ALTER GROUPBUFFERPOOL command to lower the GBPCHKPT GBP checkpoint frequency.
- Use the MVS SETXCF START,REBUILD command to expand the GBP coupling facility structure. If you rebuild, you can use the ALTER GROUPBUFFERPOOL command to lower the RATIO so that a higher percentage of the GBP storage is given to data entries over directory entries.
- If you are running on MVS 5.2 or a subsequent release, you can use the MVS SETXCF START,ALTER command to expand the size of the GBP (the maximum size to which you can expand is determined by the SIZE parameter in the CFRM policy that was used to allocate the GBP).

If you don't do anything to relieve the storage shortage, message DSNB325A appears if the GBP reaches 90 percent full.

Message DSNB327I appears when the GBP storage shortage is relieved.

You can monitor the amount of GBP storage in use with the DISPLAY GBPOOL command and with the DB2 statistics trace. Monitor the castout activity to ensure that castouts are keeping up with the rate of GBP-write activity. You can use Performance trace

classes 20 and 21 (specifically, IFCIDs 0261, 0262 and 0263) to do more detailed monitoring of the DB2 castout processing.

DSNB320I *csect-name* **- DBNAME** *database* **SPACE NAME** *spacename* **PARTITION** *part-number* **IN GROUP BUFFERPOOL** *gbpname* **IS IN GROUP BUFFERPOOL RECOVERY PENDING STATE**

Explanation: This message reports the group buffer pool recovery pending (GRECP) status of the named table space or index space partition.

Severity: 4 (warning)

System action: The data in the named partition is not accessible.

User response: Issue the -START DATABASE command on the partition to remove the GRECP status and make the data accessible again. You can also use the RECOVER or LOAD utility with the REPLACE option to clear the GRECP status.

DSNB321I *csect-name* **- DBNAME** *database* **SPACE NAME** *spacename* **IN GROUP BUFFERPOOL** *gbpname* **IS IN GROUP BUFFERPOOL RECOVERY PENDING STATE**

Explanation: This message reports the group buffer pool recovery pending (GRECP) status of the named table space or index space.

Severity: 4 (warning)

System action: The table space or index space is not accessible.

User response: Issue the -START DATABASE command on the table space or index space to remove the GRECP status and make the data accessible. You can also use the RECOVER or LOAD utility with the REPLACE option to clear the GRECP status.

DSNB322I *csect-name* **- TABLE SPACE OR INDEX SPACE WITH DBID X'***dbid***' PSID X'***psid***' PARTITION** *part-number* **IN GROUP BUFFERPOOL** *gbpname* **IS IN GROUP BUFFERPOOL RECOVERY PENDING STATE AND CANNOT BE AUTOMATICALLY RECOVERED**

Explanation: This message reports the group buffer pool recovery pending (GRECP) status of the table space or index space partition identified by the internal identifiers DBID and PSID. This message is issued instead of message DSNB320I when the name of the table space or index space is not available. The partition cannot be automatically recovered because there are retained physical locks (P-locks) on the partition.

Severity: 4 (warning)

System action: The data in the identified partition is not accessible.

User response: Issue the -DISPLAY DATABASE(*) SPACENAM(*) RESTRICT command to determine the table spaces or index spaces partition in GRECP status. First, start the failed DB2 members that hold retained P-locks on the partition.Issue the -START DATABASE command on the partition to remove the GRECP status and make the data accessible again. You can also use the RECOVER or LOAD utility with the REPLACE option to clear the GRECP status.

DSNB323I *csect-name* **- TABLE SPACE OR INDEX SPACE WITH DBID X'***dbid***' PSID X'***psid***' IN GROUP BUFFERPOOL** *gbpname* **IS IN GROUP BUFFERPOOL RECOVERY PENDING STATE AND CANNOT BE AUTOMATICALLY RECOVERED**

Explanation: This message reports the group buffer pool recovery pending (GRECP) status of the table space or index space identified by the internal identifiers DBID and PSID. This message is issued instead of message DSNB321I when the name of the table space or index space is not available. The table space or index space cannot be automatically recovered because there are retained physical locks (P-locks) on the space.

Severity: 4 (warning)

System action: The table space or index space is not accessible.

User response: Issue the -DISPLAY DATABASE(*) SPACENAM(*) RESTRICT command to determine the table spaces or index spaces in GRECP status. First, start the failed DB2 members that hold retained P-locks on the table space or index space.Issue the -START DATABASE command on the table space or index space to remove the GRECP status and make the data accessible again. You can also use the RECOVER or LOAD utility with the REPLACE option to clear the GRECP status.

DSNB325A *csect-name* **THERE IS A CRITICAL SHORTAGE OF SPACE IN GROUP BUFFER POOL** *gbp-name*

Explanation: The amount of storage in the group buffer pool (GBP) coupling facility structure available for writing new pages is critically low. The GBP is 90 percent full with changed data. As the GBP fills with changed data, the castout thresholds are continuously reached, and the castout engines must continuously work to attempt to keep the number of changed pages in the GBP below the castout thresholds. In addition, the more the GBP is filled with changed data, the less efficient are the coupling facility storage management algorithms. Completely running out of GBP storage can

result in transaction delays and in pages being added to the logical page list (LPL).

This message stays on the MVS console and is highlighted.

System action: DB2 processing continues.

This message might remain highlighted and might not be automatically deleted in a pre-SP5.2.1-JES3 environment.

Operator response: Contact the system programmer to fix the storage problem.

System programmer response: You can relieve the GBP storage shortage by taking one or more of the following actions:

- Use the ALTER GROUPBUFFERPOOL command to lower the CLASST and GBPOOLT castout thresholds.
- Use the ALTER GROUPBUFFERPOOL command to lower the GBPCHKPT GBP checkpoint frequency.
- Use the MVS SETXCF START,REBUILD command to expand the GBP coupling facility structure. If you rebuild, you can use the ALTER GROUPBUFFERPOOL command to lower the RATIO so that a higher percentage of the GBP storage is given to data entries over directory entries.
- If you are running on MVS 5.2 or a subsequent release, you can use the MVS SETXCF START,ALTER command to expand the size of the GBP (the maximum size to which you can expand is determined by the SIZE parameter in the CFRM policy that was used to allocate the GBP).

If you don't do anything to relieve the storage shortage, message DSNB228I appears if a write to the GBP fails due to a 'GBP full' condition. If DB2 detects a 'full' condition, it will suspend the transaction, initiate castout activity, and then retry the write to the GBP. If the GBP write still fails after several retry attempts, then DB2 adds the page to LPL.

Message DSNB327I appears when the GBP storage shortage is relieved.

You can monitor the amount of GBP storage in use with the DISPLAY GBPOOL command and with the DB2 statistics trace. Monitor the castout activity to ensure that castouts are keeping up with the rate of GBP-write activity. You can use Performance trace classes 20 and 21 (specifically, IFCIDs 0261, 0262 and 0263) to do more detailed monitoring of the DB2 castout processing.

DSNB327I *csect-name* **GROUP BUFFER POOL** *gbp-name* **HAS ADEQUATE FREE SPACE**

Explanation: The group buffer pool (GBP) coupling facility structure has adequate space available for the writing of new pages to the GBP.

This message may be issued when either of the following occurs:

- Rebuilding of GBP reduces changed page count to 60% or less of GBP capacity.
- 'changed data.' entries decrease to 60% or lower when they were previously 75% GBP capacity or higher.

System action: DB2 processing continues.

DSNB330I *csect-name* **GROUP BUFFER POOL** *gbpname* **CANNOT BE CONNECTED FOR REBUILD DB2 REASON CODE =** *reason1* **MVS IXLCONN REASON CODE =** *reason2*

Explanation: DB2 was not able to connect to the named group buffer pool for rebuild. *reason1* is the DB2 reason code. *reason2* is the MVS IXLCONN reason code. See *MVS/ESA Programming: Sysplex Services Reference* for an explanation of MVS IXLxxxx reason codes.

System action: DB2 stops the rebuild.

Problem determination: If the MVS IXLCONN reason code is xxxx0C08 (structure allocation failed because there was no suitable facility to allocate based on the preference list in the MVS CFRM administrative policy), or if the MVS IXLCONN reason code is xxxx084C (user does not have proper Security Access Facility (SAF) authorization), then IFCID 0250, if active, contains additional diagnostic information.

DSNB331I *csect-name* **REBUILD STARTED FOR GROUP BUFFER POOL** *gbpname* **REASON =** *reason*

Explanation: A rebuild has been initiated for the named group buffer pool. *reason* indicates the reason why the rebuild was initiated. *reason* can be one of the following:

OPERATOR
The operator initiated a SETXCF command to initiate the rebuild.

LOSSCONN
MVS detected a loss of connectivity to the group buffer pool.

This message is issued by each DB2 member participating in the rebuild.

System action: The DB2 members that are actively connected to the named group buffer pool participate in the rebuild. DB2 accomplishes the group buffer pool rebuild processing in two phases:

1. Quiesce phase. Each member quiesces all local access to the group buffer pool.
2. Connect phase. Each member connects to the new structure. Each page set or partition castout owner that has connectivity to original structure is responsible for writing pages from the original

structure to the new one. If there are not enough data entries in the new structure, the page set and partition owners cast those pages out to DASD instead. Each member issues message DSNB332I after it completes this activity. After all members have completed their castout owner rebuild work, the group buffer pool structure owner does the final sweep and checks to see if there are any changed pages still remaining in the original structure. If so, the structure owner reads the changed pages from the original structure and writes them to the new one. The structure owner issues message DSNB333I after it has completed the final sweep.

After all members have responded to MVS indicating that its connect phase processing is complete, MVS then notifies each member that the rebuild is complete. Each DB2 member issues message DSNB338I indicating that the rebuild has completed successfully.

If an error occurs during the rebuild, the rebuild is stopped. Each DB2 member issues message DSNB335I indicating the reason why the rebuild is stopping, followed by message DSNB339I indicating that the rebuild has not completed successfully. The operator can also stop rebuilds using the MVS SETXCF command.

DSNB332I *csect-name* **THIS MEMBER HAS COMPLETED CASTOUT OWNER WORK FOR GROUP BUFFER POOL** *gbpname* **PAGES CAST OUT FROM ORIGINAL STRUCTURE =** *value1* **PAGES WRITTEN TO NEW STRUCTURE =** *value2*

Explanation: This DB2 member has completed its part of the rebuild processing for the named group buffer pool for the page sets and partitions for which it is the castout owner.

value1 indicates the number of pages that this member has castout to DASD for the rebuild. If the new structure has a sufficient number of data entries, then *value1* should normally be zero because DB2 writes the changed pages from the original structure to the new structure. However, if the new structure does not have sufficient space, then *value1* could be greater than zero because DB2 responds to insufficient space in the new structure by casting the changed pages in the original structure to DASD instead of writing them to the new structure.

value2 indicates how many changed pages this member wrote to the new structure from the original as part of the rebuild process. If the new structure has sufficient space, then each DB2 that has connectivity to the original structure writes the changed pages for page sets and partitions for which it is the castout owner from the original structure to the new structure, and *value2* indicates the number of pages that the DB2 member wrote.

value1 and *value2* would be zero if this member has no page set or partition castout ownership responsibility.

This message is issued by each DB2 member that has connectivity to the original group buffer pool, and it is always be preceded by message DSNB331I or DSNB740I.

System action: If this DB2 member is not the group buffer pool structure owner, then its rebuild work is done, and it notifies MVS that it has completed its part of the rebuild and awaits later notification that all members have completed. If this member is the structure owner, then it proceeds to its final sweep processing (see DSNB333I).

Refer to "System Action" for message DSNB331I for an explanation of the group buffer pool rebuild processing.

DSNB333I *csect-name* **FINAL SWEEP COMPLETED FOR GROUP BUFFER POOL** *gbpname* **PAGES WRITTEN TO NEW STRUCTURE =** *value1*

Explanation: DB2 has completed the final sweep for the rebuild of the named group buffer pool. A final sweep means that the DB2 member that is the group buffer pool structure owner ensures that all changed pages have either been cast out to DASD or have been written to the new structure after all the page set and partition owners have finished their work. If any changed pages are found in the original structure during the final sweep, then the structure owner writes them out to the new structure. *value1* should be zero in most cases for a manual rebuild. *value1* will probably be greater than zero in most cases where the rebuild was initiated because a DB2 lost connectivity to the group buffer pool.

This message is issued by the structure owner and is always preceded by message DSNB332I.

System action: Immediately after issuing this message, the structure owner notifies MVS that the rebuild has completed, and MVS should soon begin notifying each participating member that the rebuild is complete.

Refer to "System Action" for message DSNB331I for an explanation of the group buffer pool rebuild processing.

DSNB335I *csect-name* **REBUILD STOPPED FOR GROUP BUFFER POOL** *gbpname* **REASON =** *reason1* **DB2 REASON CODE =** *reason2*

Explanation: A rebuild has been stopped for the named group buffer pool.

reason1 indicates why the rebuild was stopped:

OPERATOR
The operator initiated an MVS SETXCF command to stop the rebuild.

LOSSCONNNEW
One or more of the DB2 members lost connectivity to the new group buffer pool structure while the rebuild was in progress.

LOSSCONNOLD
One or more of the DB2 members lost connectivity to the original group buffer pool structure while the rebuild was in progress.

STRFAILNEW
The new group buffer pool structure failed while the rebuild was in progress.

STRFAILOLD
The original group buffer pool structure failed while the rebuild was in progress.

DB2REASON
DB2 decided to stop the rebuild. The reason code given in *reason2* indicates why DB2 stopped the rebuild.

INSUFFCONN
New structure does not provide connectivity that is better than or equivalent to the current structure.

NOBETRCONN
New structure does not provide better connectivity than the current structure for a *LossConn* rebuild

If *reason1* indicates 'DB2REASON', then *reason2* specifies a DB2 reason code that indicates why DB2 stopped the rebuild. If *reason1* is not 'DB2REASON', then *reason2* is zero.

This message is normally preceded by message DSNB331I, which indicates that the rebuild was started and the reason why the rebuild was started. This message is issued by each DB2 member participating in the rebuild. This message is followed by message DSNB339I.

System action: The new structure is discarded. If the original structure is still usable, then all members that still have connectivity to the original structure resume using it. Members that do not have connectivity to the original structure disconnect, and their connections become failed-persistent.

If the original structure is not usable, then DB2 initiates damage assessment processing for the group buffer pool.

System programmer response: If the rebuild stopped because of "DB2REASON", see the explanation for the reason code given.

DSNB338I *csect-name* **THE REBUILD FOR GROUP BUFFER POOL** *gbpname* **HAS COMPLETED SUCCESSFULLY**

Explanation: The rebuild for the named group buffer pool has successfully completed.

This message is preceded by message DSNB331I which indicates the reason why the rebuild was started. This message is issued by each DB2 member participating in the rebuild.

System action: The original structure is discarded and all DB2s that are connected to the group buffer pool begin using the new structure.

DSNB339I *csect-name* **THE STOP OF THE REBUILD FOR GROUP BUFFER POOL** *gbpname* **HAS COMPLETED**

Explanation: The stop of the rebuild for the named group buffer pool has completed.

This message is preceded by message DSNB335I which indicates the reason why the rebuild was stopped. This message is issued by each DB2 member participating in the rebuild.

System action: The new structure is discarded. If the original structure is still usable, then all members that still have connectivity to the original structure resume using it, and members that do not have connectivity to the original structure disconnect and their connection goes into a failed-persistent state.

If the original structure is not usable, then DB2 initiates damage assessment processing for the group buffer pool.

DSNB340I *csect-name* **UNEXPECTED ERROR DURING REBUILD FOR GROUP BUFFER POOL** *gbpname* **MVS** *service-name* **RETURN CODE =** *rc* **MVS** *service-name* **REASON CODE =** *reason*

Explanation: DB2 received an unexpected error from MVS during rebuild processing for the named group buffer pool.

service-name is one of the following:
IXLREBLD
IXLEERSP

rc is the return code from the MVS service and *reason* is the reason code.

System action: DB2 stops the rebuild.

System programmer response: See *MVS/ESA Programming: Sysplex Services Reference* for an explanation of the return and reason code for the named MVS service.

DSNB350I *csect-name* **DBNAME** *database* **SPACE NAME** *spacename* **CANNOT BE AUTOMATICALLY RECOVERED. DB2 REASON CODE =** *reason*

Explanation: Following a group buffer pool failure, DB2 cannot automatically recover the named table space or index space. *reason* specifies a DB2 reason code that indicates why DB2 could not initiate the recovery.

Severity: 4 (warning)

System action: The table space or index space remains in group buffer pool recovery pending (GRECP) status and is inaccessible.

User response: Use the DB2 -START DATABASE command on the table space or index space to remove the GRECP status and make the data accessible. You can also use the RECOVER or LOAD utility with the REPLACE option to clear the GRECP status.

DSNB351I *csect-name* **DBNAME** *database* **SPACE NAME** *spacename* **PARTITION** *part#* **CANNOT BE AUTOMATICALLY RECOVERED. DB2 REASON CODE =** *reason*

Explanation: Following a group buffer pool failure, DB2 cannot automatically recover the named table space or index space partition. *reason* specifies a DB2 reason code that indicates why DB2 could not initiate the recovery.

Severity: 4 (warning)

System action: The partition remains in group buffer pool recovery pending (GRECP) status and is inaccessible.

User response: Use the DB2 -START DATABASE command on the partition to remove the GRECP status and make the data accessible. You can also use the RECOVER or LOAD utility with the REPLACE option to clear the GRECP status.

DSNB352I *csect-name* **AUTOMATIC RECOVERY CANNOT BE INITIATED FOR GROUP BUFFER POOL** *gbp-name* **REASON =** *reason*

Explanation: Following a failure of the named group buffer pool, DB2 is not able to initiate automatic recovery.

reason indicates the reason why automatic recovery was not initiated:

AUTORECNO
The AUTOREC option is set to NO for the group buffer pool.

CONNRELEASE
At least one actively connected DB2 member is not at the proper release to support automatic recovery.

Severity: 4 (warning)

System action: The page sets and partitions that are in group buffer pool recovery pending (GRECP) status remain inaccessible.

User response: Use the DB2 -START DATABASE command on the page sets and partitions to remove the GRECP status and make the data accessible. You can also use the RECOVER or LOAD utility with the REPLACE option to clear the GRECP status.

If *reason* is AUTORECNO and you want to enable automatic recovery for the group buffer pool, use the ALTER GROUPBUFFERPOOL command to set AUTOREC to YES for the group buffer pool.

If *reason* is CONNRELEASE, then you must ensure that all DB2 members in the group are running at the appropriate release. Automatic recovery is not supported unless all members are running Version 5 or a later release.

DSNB353I *csect-name* **ASYNCHRONOUS GRECP RECOVERY IN PROGRESS FOR GROUP BUFFER POOL** *bpoolname*

Explanation: Following a group buffer pool failure, DB2 triggers an automatic recovery process to attempt to recover GRECP page sets for the specified group buffer pool. This message is issued once by any DB2 member who is the page set castout owner on one of the objects being recovered.

System action: Check for the DSNB354I message to verify the completion of the automatic recovery process.

DSNB354I *csect-name* **ASYNCHRONOUS GRECP RECOVERY HAS COMPLETED FOR GROUP BUFFER POOL** *bpoolname***.**

Explanation: DB2 has completed the GRECP recovery process for those GRECP page sets in the group buffer pool specified in the message. This message is issued once by any DB2 member who previously issued a DSNB353I message, who is the page set castout owner on one of the objects being recovered.

DSNB357I *csect-name* **AUTOMATIC LPL RECOVERY SUPPRESSED FOR OBJECT TYPE** *object-type* **OBJECT NAME** *object-name* **REASON** *reason-type*

| **Explanation:** One or more pages in the object that is identified in the message have been added to the logical page list (LPL). However, DB2 cannot initiate automatic LPL recovery due to an environmental condition that prevents automatic LPL recovery.

The object could be a table space, an index space, or a partition within a table space or an index space that is identified by the *object-type*. *object-name* identifies the name.

If the object is a table space or an index space, the format of the name is *dbname.spacename*. If the object is a partition within a table space or an index space, the format of the name is *dbname.spacename.partnumber*. *dbname* is the database name. *spacename* is the table space or the index space name. *partnumber* is the partition number.

reason indicates the condition that prevents DB2 from being able to initiate automatic LPL recovery. Here is a list of the potential values for *reason*:

DASD DB2 encountered a DASD I/O error.

RESTART
DB2 is in Restart or End_Restart mode.

GBP DB2 Group Buffer Pool structure failure or 100% loss of connectivity.

System action: The Automatic LPL Recovery is not initiated.

Programmer response: Check the reason that is specified to solve the system condition. To recover the LPL page, perform one of the following actions:

- Issue the START DATABASE command with the SPACENAM option
- Run the RECOVER or LOAD utility with REPLACE keyword.

DSNB401I BUFFERPOOL NAME *bpname*, **BUFFERPOOL ID** *bpid*, **USE COUNT** *ucount*

Explanation: This message contains the display information that results from a DISPLAY BUFFERPOOL command. For each buffer pool, this message is the first of a set of messages that display information from the DISPLAY BUFFERPOOL command.

The statistics are:

bpname
The buffer pool external name (BP0, BP1,..., BP49, BP8KB, BP16KB BP32K, BP32K1,..., BP32K9).

bpid The buffer pool internal identifier (0-49, 80-89).

ucount The number of open table spaces or index spaces that refer to this buffer pool. A zero use count means that the buffer pool is inactive.

System action: Processing continues.

DSNB402I BUFFERPOOL SIZE = *vpsize* **BUFFERS ALLOCATED =** *vpalc* **TO BE DELETED =** *vptbd* **IN-USE/UPDATED =** *vpcba* **BUFFERS ACTIVE =** *vpactive*

Explanation: This message, displayed by the DISPLAY BUFFERPOOL command, describes the allocation status of a buffer pool. This message is always preceded by message DSNB401I identifying the associated buffer pool.

The statistics are:

vpsize The user-specified buffer pool size.

vpalc Number of allocated buffers in an active buffer pool.

vptbd Number of buffers to be deleted in an active buffer pool because of pool contraction.

vpcba Number of currently active (that is, not stealable) buffers in the buffer pool.

vpactive
Number of currently active (that is, contain data) buffers in the buffer pool. For a page fixed (PGFIX) set to YES, it also indicates the number of buffers that are fixed in real storage. Unless the buffer pool is over-allocated, this value should be equal to *vpalc* in a steady state.

System action: Processing continues.

DSNB404I THRESHOLDS — VP SEQUENTIAL = *vpseq* **DEFERRED WRITE =** *dwt* **VERTICAL DEFERRED WRT =** *dwv1*,*dwv2* **PARALLEL SEQUENTIAL =** *vppseqt* **ASSISTING PARALLEL SEQT =** *vpxpseqt*

Explanation: This message is displayed by the DISPLAY BUFFERPOOL command to describe the user-modifiable thresholds for the buffer pool. This message is always preceded by message DSNB401I, which identifies the associated buffer pool.

The input threshold variables are:

vpseq Sequential steal threshold for the virtual pool, expressed as a percentage of the total buffer pool size.

dwt Free buffer deferred write threshold for the buffer pool, expressed as a percentage of the total buffer pool size.

dwv1 Vertical deferred write threshold for the buffer pool, expressed as a percentage of the total buffer pool size.

dwv2 Vertical deferred write threshold for the buffer pool, expressed as an absolute number of buffers. *dwv2* is used to determine the

threshold only if *dwv1* is zero and *dwv2* is non-zero; otherwise *dwv1* is used to determine the threshold.

vppseqt Sequential threshold for parallel query processing, expressed as a percentage of the virtual sequential steal threshold. When the threshold is set to 0, there is no parallel query processing support.

vpxpseqt Assisting parallel sequential threshold, expressed as a percentage of the sequential threshold for parallel query processing. *vpxpseqt* allows you to control how much buffer resource is used when this DB2 is assisting another member of the group in parallel processing. When the threshold is set to 0, this buffer pool will not be used to assist other data sharing members in processing a query. In a non-data-sharing environment, this value is ignored.

System action: Processing continues.

DSNB406I PGFIX ATTRIBUTE - CURRENT = *current-pgfix* PENDING = *pending-pgfix* PAGE STEALING METHOD = *pgsteal*

Explanation: This message, displayed by the DISPLAY BUFFERPOOL command, describes the current and pending PGFIX and the currently used PGSTEAL attributes. This message is always preceded by message DSNB401I identifying the associated buffer pool.

current-pgfix is the buffer pool page fix (PGFIX) attribute that is applicable to the current allocation of the buffer pool.

YES The buffer pool is page fixed in real storage.

NO The buffer pool is not page fixed in real storage.

pending-pgfix is the pending buffer pool PGFIX attribute that will be applied for the next allocation of the virtual buffer pool.

YES The buffer pool is page fixed in real storage.

NO The buffer pool is not page fixed in real storage.

pgsteal is the page stealing method (PGSTEAL) that is in use for the buffer pool.

LRU Specifies the buffer pool buffers are managed using the least recently used (LRU) algorithm. This is the default.

FIFO Specifies the buffer pool buffers are managed using the first in first out (FIFO) algorithm.

System action: Processing continues.

DSNB408I NO DETAIL STATISTICS AVAILABLE FOR BUFFER POOL *bpname*

Explanation: This message is displayed by the DISPLAY BUFFERPOOL command with the DETAIL option specified. It indicates that no detail statistics are available for the requested buffer pool because the pool was not activated since DB2 startup.

System action: Processing continues.

DSNB409I INCREMENTAL STATISTICS SINCE *basetime*

Explanation: This message is displayed by the DISPLAY BUFFERPOOL command with the DETAIL(INTERVAL) option specified. It indicates the start time of the interval for which the statistics were accumulated (that is, either the time of the previous incremental display or the time the buffer pool was first activated, if there was no previous incremental display).

System action: Processing continues.

DSNB410I CUMULATIVE STATISTICS SINCE *basetime*

Explanation: This message is displayed by the DISPLAY BUFFERPOOL command with the DETAIL(*) option specified. It indicates the start time of the interval for which the statistics were accumulated (that is, the time the buffer pool was first activated).

System action: Processing continues.

DSNB411I RANDOM GETPAGE = *rgp* SYNC READ I/O (R) = *srr* SEQ. GETPAGE = *sgp* SYNC READ I/O (S) = *srs* DMTH HIT = *dmt* PAGE-INS REQ = *pir*

Explanation: This message is displayed by the DISPLAY BUFFERPOOL command with the DETAIL option specified. It lists the page read statistics for the buffer pool. This message is always preceded by message DSNB401I identifying the associated buffer pool.

The statistics are:

rgp Number of nonsequential GETPAGE requests

srr Number of synchronous read I/O operations for nonsequential GETPAGE requests

sgp Number of sequential GETPAGE requests

srs Number of synchronous read I/O operations for sequential GETPAGE requests

dmt Number of times data management threshold was hit

pir Number of page-ins required for read I/O

System action: Processing continues.

DSNB412I SEQUENTIAL PREFETCH - REQUESTS = *pft* PREFETCH I/O = *pio* PAGES READ = *pfp*

Explanation: This message is displayed by the DISPLAY BUFFERPOOL command with the DETAIL option specified. It lists the sequential prefetch statistics for the buffer pool. This message is always preceded by message DSNB401I identifying the associated buffer pool.

The statistics are:

pft Number of times sequential prefetch was requested

pio Number of sequential prefetch read I/O operations

pfp Number of pages read because of sequential prefetch

System action: Processing continues.

DSNB413I LIST PREFETCH - REQUESTS = *pft* PREFETCH I/O = *pio* PAGES READ = *pfp*

Explanation: This message is displayed by the DISPLAY BUFFERPOOL command with the DETAIL option specified. It shows the list prefetch statistics for the buffer pool. This message is always preceded by message DSNB401I identifying the associated buffer pool.

The statistics are:

pft Number of times list prefetch was requested

pio Number of list prefetch read I/O operations

pfp Number of pages read because of list prefetch

System action: Processing continues.

DSNB414I DYNAMIC PREFETCH - REQUESTS = *pft* PREFETCH I/O = *pio* PAGES READ = *pfp*

Explanation: This message is displayed by the DISPLAY BUFFERPOOL command with the DETAIL option specified. It lists the dynamic prefetch statistics for the buffer pool. This message is always preceded by message DSNB401I identifying the associated buffer pool.

The statistics are:

pft Number of times dynamic prefetch was requested

pio Number of dynamic prefetch read I/O operations

pfp Number of pages read because of dynamic prefetch

System action: Processing continues.

DSNB415I PREFETCH DISABLED - NO BUFFER = *pfd* NO READ ENGINE = *ree*

Explanation: This message is displayed by the DISPLAY BUFFERPOOL command with the DETAIL option specified. It lists prefetch statistics for the buffer pool. This message is always preceded by message DSNB401I identifying the associated buffer pool.

The statistics are:

pfd Number of times prefetch was disabled (no buffer)

ree Number of times prefetch was disabled (no asynchronous read processor available)

System action: Processing continues.

DSNB420I SYS PAGE UPDATES = *sws* SYS PAGES WRITTEN = *pws* ASYNC WRITE I/O = *wio* SYNC WRITE I/O = *imw* PAGE-INS REQ = *pir*

Explanation: This message is displayed by the DISPLAY BUFFERPOOL command with the DETAIL option specified. It lists page write statistics for the buffer pool. This message is always preceded by message DSNB401I identifying the associated buffer pool.

The statistics are:

sws Number of system page updates

pws Number of system pages written

wio Number of asynchronous write I/O operations

imw Number of synchronous write I/O operations

pir Number of page-ins required for write I/O

System action: Processing continues.

DSNB421I DWT HIT= *dwt* VERTICAL DWT HIT=*vdw* NO WRITE ENGINE=*wee*

Explanation: This message is displayed by the DISPLAY BUFFERPOOL command with the DETAIL option specified. It lists page-write statistics for the buffer pool. This message is always preceded by message DSNB401I identifying the associated buffer pool.

The statistics are:

dwt Number of times deferred write threshold was hit

vdw Number of times vertical deferred write threshold was hit

wee Number of times write engine (asynchronous write processor) was not available for I/O operations

System action: Processing continues.

DSNB440I PARALLEL ACTIVITY — PARALLEL REQUEST = *tpa* DEGRADED PARALLEL = *dpa*

Explanation: This message is issued in response to the DISPLAY BUFFERPOOL command with DETAIL specified. It lists statistics about parallel activities. This message is always preceded by message DSNB401I identifying the associated buffer pool.

The variables are:

tpa Total number of negotiations with the buffer pool for getting the requested number of sequential prefetch streams

dpa Total number of times the negotiation resulted in the degraded mode of parallel operations

DSNB453I VP CACHED PAGES - CURRENT = *vcount* MAX = *mvcount* CHANGED = *ccount* MAX = *mccount*

Explanation: This message might be displayed for an active buffer pool by the DISPLAY BUFFERPOOL command with LSTATS specified. It lists the number of cached pages and the number of changed pages in the buffer pool for a data set. The message is only issued if there are nonzero values to display. This message is always preceded by message DSNB401I identifying the associated buffer pool, DSNB464I or DSNB465I identifying the table space or index space, and DSNB466I identifying the data set.

The statistics are:

vcount Number of cached pages in VP for the data set (that is, the number of buffers that contain pages for the data set in the buffer pool).

mvcount
Maximum number of cached pages in VP for the data set since the last DISPLAY BUFFERPOOL command with the LSTATS option was issued.

ccount Number of changed pages in VP for the data set (that is, the number of buffers that were changed in the buffer pool for the data set).

mccount
Maximum number of changed pages in VP for the data set since the last DISPLAY BUFFERPOOL command with the LSTATS option was issued.

This message is issued by the following CSECT: DSNB1DBP

System action: Processing continues.

DSNB455I SYNCHRONOUS I/O DELAYS — AVERAGE DELAY = *avd* MAXIMUM DELAY = *mxd* TOTAL PAGES = *tpg*

Explanation: This message might be displayed for an active buffer pool by the DISPLAY BUFFERPOOL command with the LSTATS option specified. It lists synchronous I/O delay statistics for a data set belonging to an open table space or index space associated with the buffer pool. The values listed are incremental since the last display for the data set. The message is only issued if there are nonzero values to display. This message is always preceded by message DSNB401I identifying the associated buffer pool, DSNB464I or DSNB465I identifying the table space or index space, and DSNB466I identifying the relative data set within the table space or index space.

The statistics are:

avd Average I/O delay in milliseconds for pages in the data set

mxd Maximum I/O delay in milliseconds for pages in the data set

tpg Total number of pages read or written for the data set

System action: Processing continues.

DSNB456I ASYNCHRONOUS I/O DELAYS — AVERAGE DELAY = *avd* MAXIMUM DELAY = *mxd* TOTAL PAGES = *tpg* TOTAL I/O COUNT = *tio*

Explanation: This message might be displayed for an active buffer pool by the DISPLAY BUFFERPOOL command with the LSTATS option specified. It lists asynchronous I/O delay statistics for a data set belonging to an open table space or index space associated with the buffer pool. The values listed are incremental since the last display for the data set. The message is only issued if there are nonzero values to display. This message is always preceded by message DSNB401I identifying the associated buffer pool, DSNB464I or DSNB465I identifying the table space or index space, and DSNB466I identifying the relative data set within the table space or index space.

The statistics are:

avd Average I/O delay in milliseconds for pages in the data set

mxd Maximum I/O delay in milliseconds for pages in the data set

tpg Total number of pages read or written for the data set

tio Total number of I/Os issued for the data set

System action: Processing continues.

DSNB459I *csect-name* **OPEN FAILURE HAS OCCURRED FOR DATA SET** *dsn*

Explanation: This message might be displayed for an active buffer pool by the DISPLAY BUFFERPOOL command with the LIST option specified. It indicates that a previous attempt to access a data set (identified by its relative data set number within a table space or index space) failed due to an allocation or open error.

This message is always preceded by a DSNB401I message identifying the associated buffer pool and a DSNB464I or DSNB465I message identifying the table space or index space.

System action: Processing continues.

DSNB460I **--PAGE SET/PARTITION LIST INFORMATION----DATA SHARING INFO-- TS GBP MEMBER CASTOUT USE P-LOCK DATABASE SPACE NAME PART IX DEP NAME OWNER COUNT STATE**

Explanation: This message displays the column headings that indicate the beginning of a multiple line display generated by a DISPLAY BUFFERPOOL command. The format of the multiple line display has three parts:

- The column headers (DSNB460I), which are underlined
- Detailed space information for all spaces and partitions requested by the command (DSNB462I)
- An ending message (DSNB463I) that might be displayed if no spaces matched the command criteria

The text of message DSNB462I provides the detail lines that follow the headings generated by this message. If the DISPLAY BUFFERPOOL LIST command finds more than 255 lines of output to display, the report will be broken up and new headings (DSNB460I) will be displayed with '(CONTINUED)' showing in the heading text. This is needed due to the MVS limitation of 255 messages for multi-line WTO messages.

System action: Processing continues.

DSNB461I

Explanation: This message appears within a multiple line display generated by a DISPLAY BUFFERPOOL command. It follows message DSNB460I and is used to underline the column headings that are displayed by DSNB460I.

System action: Processing continues.

DSNB462I *database spacenam part ty gbp mbr castout use p-lock*

Explanation: This message displays the detail lines generated by a DISPLAY BUFFERPOOL command. It follows DSNB460I.

The detail lines contain the following information (for report readability, the fields for DBNAME and SPACE NAME might be blank. When either field is blank, one of the prior detail lines contains the suppressed value).

database Name of the database that is associated with the page set/partition.

spacenam Name of the table space or index space.

part The partition number. It is blank for a simple table space or simple index space. For non partitioned indexes on a partitioned table space, it is the logical partition number preceded by the character L (for example, L01).

ty Whether the object is a table space (TS) or an index space (IX).

gbp Whether the page set/partition is GBP-dependent ('Y' for yes; 'N' for no).

mbr Name of the member that the detail line pertains to.

castout Whether the member identified in this line is the castout owner ('Y' for yes; blank for no).

use Number of active claimers or drainers for the page set/partition for the member displayed on this detail line.

p-lock P-lock state that the member identified in this line currently holds. Values are:

IS R/O interest. There are other members who have interest in this page set/partition. The page set/partition. is GBP-dependent.

IX R/W interest. There are other members who have interest in this page set/partition. The page set/partition is GBP-dependent.

S R/O interest. There may be other members reading the page set/partition. The page set/partition is not GBP-dependent.

SIX R/W interest. There may be other members who are reading the page set/partition. The page set/partition is GBP-dependent.

NSU R/W interest. The page set/partition is GBP-dependent.

X R/W interest. There are no other

members accessing the page set/partition. The page set/partition is not GBP-dependent.

US A temporary state that can be held by a restarting DB2 when 'waiting for retained locks' is enabled.

A number that is to be used as a diagnostic aid. A number is displayed only when a DISPLAY BUFFERPOOL command encounters a lock state other than any of those defined above.

System action: Processing continues.

DSNB463I * * * NO OBJECTS MATCHED LIST/LSTATS SELECTION CRITERIA

Explanation: This message appears as part of the DISPLAY BUFFERPOOL output if the command did not find any page sets/partitions that matched the selection criterion. For example, the message would be displayed for DIS BPOOL(BP0) GBPDEP(Y) if the command did not find any page sets/partitions that were GBP-dependent.

User response: If you expected to see a list of page sets/partitions, ensure that the syntax of the command was coded correctly. Consider adding or changing the filter keywords to obtain a list of page sets/partitions.

System action: The command terminates normally.

DSNB464I TS USE DATABASE SPACE NAME PART IX COUNT ======== ========== ==== == =====

Explanation: This message displays underlined column headings that indicate the beginning of a multiple line display generated by a DISPLAY BUFFERPOOL command. The format of the multiple line display has three parts:

- Underlined column headings (DSNB464I).
- Detailed space information for the spaces and partitions that are requested by the command (DSNB465I).
- An ending message (DSNB463I) that may be displayed if no spaces matched the command criteria.

The text of message DSNB465I provides the detail lines that follow the headings generated by this message. If the DISPLAY BUFFERPOOL LIST command finds more than 255 lines of output to display, the report will be broken up and new headings (DSNB464I) will be displayed with '(CONTINUED)' showing in the heading text. This is needed due to the MVS limitation of 255 messages for multi-line WTO messages.

System action: Processing continues.

DSNB465I database spacenam part ty use

Explanation: This message displays the detail lines generated by the DISPLAY BUFFERPOOL command. It follows DSNB464I.

The detail lines contain the following information (for report readability, the fields for DBNAME and SPACE NAME might be blank. When either field is blank, one of the prior detail lines will contain the suppressed value).

database
Name of the database that is associated with the page set/partition.

spacenam
Name of the table space or index space.

part The partition number. It is blank for a simple table space or simple index space. For non partitioned indexes on a partitioned table space, it is the logical partition number preceded by the character L

ty Whether the object is a table space (TS) or an index space (IX).

System action: Processing continues.

DSNB466I ----------PAGE SET/PARTITION STATISTICS----------

Explanation: This message appears as part of the DISPLAY BUFFERPOOL output if the command was specified with the LSTATS option. It shows statistics for data sets.

System action: Processing continues.

DSNB467I STATISTICS FOR *type dbname.spacenam* **DATA SET #:** *dsn#* **USE COUNT:***ucount*

Explanation: This message might be displayed for an active buffer pool by the DISPLAY BUFFERPOOL command with the LSTATS option specified. It identifies the page set/partition type (table space/index space), database name, space name, number of claimers/drainers and the relative data set number within a table space or index space for the statistics displayed in one or more of the messages that follow this one.

type Whether the page set/partition is a table space (TABLE SPACE) or an index space (INDEX SPACE).

dbname Database name

spacenam
Name of the table space or index space.

dsn# The relative data set number within a table space or index space for the statistics displayed in one or more of the messages that follow this one.

ucount Number of applications with a claim or drain on the page set/partition.

System action: Processing continues.

DSNB468I *keyword* **KEYWORD IS IGNORED FOR NON-DATA SHARING SYSTEMS**

Explanation: The identified keyword is valid only in a data-sharing environment. Keywords GBPDEP and CASTOWNR can cause this message to be displayed.

User response: Re-enter the command without the keyword since the DB2 is a non-data-sharing DB2.

System action: The command ignores the keyword and terminates normally.

DSNB469I *keyword1* **KEYWORD IS IGNORED WHEN** *keyword2* **KEYWORD IS SPECIFIED**

Explanation: Keyword1 is incompatible with keyword2. For example, in command DIS BPOOL(BP0) GBPDEP(Y) LSTATS(*), keyword GBPDEP indicates that a list of page sets or partitions are desired (as in LIST keyword); however, keyword GBPDEP is ignored because LIST and LSTATS are two different reports.

User response: Omit keyword1 if keyword2 is used on the next entry of the command.

System action: The command ignores keyword1 and terminates normally.

DSNB499I **DISPLAY TERMINATED DUE TO INSUFFICIENT MESSAGE SPACE**

Explanation: The DISPLAY BUFFERPOOL or the DISPLAY GROUPBUFFERPOOL command was unable to obtain storage for additional messages. This should only happen for a particularly long display request, such as a detail display for a large number of buffer pools or group buffer pools.

System action: The display is truncated.

System programmer response: Reissue the -DISPLAY BUFFERPOOL or the -DISPLAY GROUPBUFFERPOOL command, specifying a smaller number of buffer pools or group buffer pools.

DSNB501I *bpname* **IS NOT A VALID BUFFER POOL NAME**

Explanation: An invalid buffer pool name was specified on an ALTER BUFFERPOOL or DISPLAY BUFFERPOOL command. THE VALID NAMES ARE BP0, BP1,..., BP49, BP32K, BP32K1,...BP32K9.

System action: The request is rejected.

System programmer response: Reissue the ALTER BUFFERPOOL or DISPLAY BUFFERPOOL command, specifying a valid buffer pool name.

DSNB502I *csect-name* **VPSIZE FOR** *bpname* **MUST BE BETWEEN** *llimit* **AND** *ulimit*

Explanation: An attempt was made to change a buffer pool's VPSIZE to an invalid value using the ALTER BUFFERPOOL command. VPSIZE is the number of buffers in the buffer pool.

bpname
Buffer pool name.

llimit Lower limit for buffer pool's VPSIZE. For BP0, the lower limit is 2000. For BP8K0, the lower limit is 1000. For BP16K0, the lower limit is 500. For all other buffer pools, the lower limit is 0.

ulimit Upper limits for buffer pool's VPSIZE. The upper limit is as follows:
- 250000000 for 4KB page-size buffer pools
- 125000000 for 8KB page-size buffer pools
- 62500000 for 16KB page-size buffer pools
- 31250000 for 32KB page-size buffer pools

System action: The request is rejected.

System programmer response: Reissue the ALTER BUFFERPOOL command, specifying a legitimate VPSIZE value.

DSNB504I **VPSEQT FOR** *bpname* **MUST BE BETWEEN 0 AND 100**

Explanation: An ALTER BUFFERPOOL command was issued to try to alter a buffer pool's VPSEQT attribute to an invalid value. VPSEQT is the buffer pool sequential steal threshold. Valid values range from 0 to 100.

System action: The request is rejected.

System programmer response: Reissue the ALTER BUFFERPOOL command, specifying a legitimate VPSEQT value.

DSNB506I **DWQT FOR** *bpname* **MUST BE BETWEEN 0 AND 90**

Explanation: An attempt was made to change a buffer pool's DWQT attribute value using the ALTER BUFFERPOOL command. DWQT is the free buffer deferred write threshold. Valid values range from 0 to 90.

System action: The request is rejected.

System programmer response: Reissue the ALTER BUFFERPOOL command, specifying a legitimate DWQT value.

DSNB507I VDWQT FOR *bpname* MUST BE BETWEEN 0 AND 90

Explanation: An attempt was made to change a buffer pool's VDWQT attribute to an invalid value using the ALTER BUFFERPOOL command. VDWQT is the vertical deferred write threshold. Valid values range from 0 to 90.

System action: The request is rejected.

System programmer response: Reissue the ALTER BUFFERPOOL command, specifying a legitimate VDWQT value.

DSNB508I *csect-name* THE TOTAL VPSIZE IS BEYOND THE MAXIMUM LIMIT

Explanation: An ALTER BUFFERPOOL command specified a VPSIZE or VPTYPE value that would cause the sum of the *ssnm*DBM1 address space storage requirements for all buffer pools to exceed 1TB. When calculating the storage requirements for the buffer pools, DB2 uses the page size (4K, 8K, 16K, or 32K) for each pool buffer.

System action: The buffer pool's VPSIZE or VPTYPE is not changed. Processing for other parameters, if requested, proceeds normally.

System programmer response: Rebalance VPSIZE for all buffer pools.

DSNB510I NO ACTION REQUESTED

Explanation: An ALTER BUFFERPOOL command was issued, but no buffer pool attributes were specified.

System action: The buffer pool's attributes are not changed.

System programmer response: Reissue the ALTER BUFFERPOOL command specifying one or more buffer pool attributes.

DSNB511I VDWQT CANNOT BE GREATER THAN DWQT

Explanation: An ALTER BUFFERPOOL command specified a DWQT and/or VDWQT value that would result in the buffer pool's VDWQT parameter being greater than its DWQT parameter.

System action: The buffer pool's VDWQT or DWQT is not changed. Processing for other parameters, if requested, proceeds normally.

System programmer response: Rebalance DWQT and VDWQT for the buffer pool.

DSNB512I *integer* IS THE CURRENT VPSIZE FOR BUFFER POOL *bpname*

Explanation: An ALTER BUFFERPOOL command specified a VPSIZE equal to the buffer pool's current buffer pool size.

System action: The buffer pool size is not changed. Processing for other parameters, if requested, proceeds normally.

DSNB514I *integer* IS THE CURRENT VPSEQT FOR BUFFER POOL *bpname*

Explanation: An ALTER BUFFERPOOL command specified an VPSEQT equal to the current sequential steal threshold of the buffer pool.

System action: The buffer pool sequential threshold is not changed. Processing for other parameters, if requested, proceeds normally.

DSNB516I *integer* IS THE CURRENT DWQT FOR BUFFER POOL *bpname*

Explanation: An ALTER BUFFERPOOL command specified a DWQT equal to the current deferred write threshold of the buffer pool.

System action: The deferred write threshold of the buffer pool is not changed. Processing for other parameters, if requested, proceeds normally.

DSNB517I *int1*,*int2* IS THE CURRENT VDWQT FOR BUFFER POOL *bpname*

Explanation: An ALTER BUFFERPOOL command specified a VDWQT equal to the current vertical deferred write threshold of the buffer pool.

System action: The vertical deferred write threshold of the buffer pool is not changed. Processing for other parameters, if requested, proceeds normally.

DSNB519I *csect-name* PGSTEAL FOR *bpname* HAS BEEN SET TO *value*.

Explanation: This message is issued in response to an ALTER BUFFERPOOL command to indicate that the requested change to the PGSTEAL parameter was accepted.

bpname is the name of the buffer pool.

value is one of the following:

LRU The buffer pool buffers are managed using the least recently used (LRU) algorithm. This is the default.

FIFO The buffer pool buffers are managed using the first in first out (FIFO) algorithm.

System action: Processing continues.

DSNB520I READY TO EXPLICITLY DELETE BUFFER POOL *bpname*

Explanation: This message is issued before draining an active buffer pool in response to an ALTER BUFFERPOOL command, which specified VPSIZE=0 against an active buffer pool.

System action: After this message is issued, DB2 quiesces all current activities against this buffer pool. DB2 locks the table space exclusively and performs physical close for all open table spaces or index spaces that reference this pool. DB2 also prevents subsequent database access that references this pool. SQL applications get an SQLCODE -904 (resource not available) with reason code 00C200E0 to indicate that the requested buffer pool size is not initialized. After the requested buffer pool is deleted, message DSNB521I is issued.

DSNB521I BUFFER POOL *bpname* **HAS BEEN EXPLICITLY DELETED**

Explanation: This message is issued after an active buffer pool is explicitly deleted with an ALTER BUFFERPOOL command. This message is always preceded by a corresponding DSNB520I message.

System action: The virtual storage and hiperspaces associated with this buffer pool are released.

DSNB522I VPSIZE FOR *bpname* **HAS BEEN SET TO** *integer*

Explanation: This message is issued in response to an ALTER BUFFERPOOL command to indicate that the requested change to the VPSIZE parameter was accepted. For an active pool, the actual expansion, contraction, or deletion does not take place until after this message is issued.

System action: Processing continues.

System programmer response: Issue the DISPLAY BUFFERPOOL command to determine the actual virtual storage usage for this pool.

DSNB524I VPSEQT FOR *bpname* **HAS BEEN SET TO** *integer*

Explanation: This message is issued in response to an ALTER BUFFERPOOL command. It indicates the successful modification of the sequential steal threshold of the buffer pool.

DSNB526I DWQT FOR *bpname* **HAS BEEN SET TO** *integer*

Explanation: This message is issued in response to an ALTER BUFFERPOOL command. It indicates the successful modification of the deferred write threshold of the buffer pool.

DSNB527I VDWQT FOR *bpname* **HAS BEEN SET TO** *int1*,*int2*

Explanation: This message is issued in response to an ALTER BUFFERPOOL command. It indicates the successful modification of the vertical deferred write threshold of the buffer pool.

DSNB529I VPPSEQT FOR *bpname* **HAS BEEN SET TO** *integer*

Explanation: This message is issued in response to the ALTER BUFFERPOOL command. It indicates the successful modification of the parallel sequential steal threshold of the buffer pool.

DSNB530I *integer* **IS THE CURRENT VPPSEQT FOR BUFFER POOL** *bpname*

Explanation: An ALTER BUFFERPOOL command specified a VPPSEQT equal to the current parallel sequential steal threshold of the buffer pool.

System action: The parallel sequential steal threshold of the buffer pool is not changed. Processing for other parameters, if requested, proceeds normally.

DSNB531I VPPSEQT FOR *bpname* **MUST BE BETWEEN 0 AND 100**

Explanation: An ALTER BUFFERPOOL command was issued to alter a buffer pool's VPPSEQT attribute to an invalid value. VPPSEQT is the parallel sequential steal threshold of the buffer pool. Valid values range from 0 to 100.

System action: The VPPSEQT value is not changed.

System programmer response: Change the VPPSEQT attribute to a valid value and reenter the command.

DSNB532I VPXPSEQT FOR *bpname* **HAS BEEN SET TO** *integer*

Explanation: This message is issued from DSNB1ABP csect in response to the ALTER BUFFERPOOL command. It indicates the successful modification of the assisting parallel sequential threshold of the buffer pool.

DSNB533I *integer* **IS THE CURRENT VPXPSEQT FOR BUFFER POOL** *bpname*

Explanation: An ALTER BUFFERPOOL command specified a VPXPSEQT equal to the current assisting parallel sequential threshold of the buffer pool.

System action: The assisting parallel sequential threshold of the buffer pool is not changed. Processing for other parameters, if requested, proceeds normally.

DSNB534I VPXPSEQT FOR *bpname* MUST BE BETWEEN 0 AND 100

Explanation: An ALTER BUFFERPOOL command was issued to change a buffer pool's VPXPSEQT attribute to an invalid value. VPXPSEQT is the assisting parallel sequential threshold of the virtual buffer pool. Valid values range from 0 to 100. This message is issued from DSNB1ABP csect.

System action: The VPXPSEQT value is not changed.

System programmer response: Change the VPXPSEQT attribute to a valid value and reenter the command.

DSNB536I *csect-name* THE TOTAL VIRTUAL BUFFER POOL STORAGE EXCEEDS THE REAL STORAGE CAPACITY

Explanation: This message is issued as a warning that the total amount of allocated virtual buffer pool storage for this DB2 subsystem has exceeded the real storage capacity of the z/OS image. The amount of real storage should be at least as much as the total amount of virtual buffer pool storage, and should also be enough to accommodate the EDM pool and DB2 working storage in addition to the virtual buffer pool storage. Paging activity in the buffers is an indication of a performance problem.

System action: Processing continues.

System programmer response: Allocate more real storage to the z/OS image. If there is not enough real storage to hold the buffers, then the number of buffers needs to be reduced. This might mean fewer concurrent users. If the number of buffers is reduced, then DB2 will issue the DSNB538I message when the total amount of virtual buffer pool storage no longer exceeds the real storage capacity.

DSNB538I *csect-name* THE TOTAL BUFFER POOL STORAGE NO LONGER EXCEEDS THE REAL STORAGE CAPACITY

Explanation: This message is issued following a previous DSNB536I message to indicate that the total amount of allocated buffer pool storage for this DB2 subsystem no longer exceeds the real storage capacity of the z/OS image.

System action: Processing continues.

DSNB539I *csect-name* AUTOLPL FOR *bpname* HAS BEEN SET TO *value*. IT WILL TAKE EFFECT ON THE NEXT ALLOCATION

Explanation: DB2 issues this message in response to an ALTER BUFFERPOOL command to ind that the requested change to the AUTOLPL parameter was accepted. The new AUTOLPL value takes effect the next time the buffer pool is allocated.

bpname is the name of the buffer pool.

value is one of the following:

YES DB2 automatically recovers pages that are in LPL status.

NO DB2 does not automatically recover pages in LPL status. To recover the LPL page, perform one of the following actions:

- Issue the START DATABASE command with the SPACENAM option
- Run the RECOVER or LOAD utility

System action: Processing continues

DSNB540I *csect-name* INSUFFICIENT LOCAL CACHE VECTOR STORAGE TO EXPAND BUFFER POOL *bpname* TO SPECIFIED SIZE *size1*. BUFFER POOL SIZE IS NOW *size2*.

Explanation: The specified buffer pool could not be expanded to the requested size (*size1*). The associated group buffer pool is connected, and the local cache vector could not be expanded to cover the requested buffer pool size.

The local cache vector is a bit array which tracks whether the page in a given buffer was modified by another system (cross-invalidated). There must be one bit in the vector for each buffer in the buffer pool. For more information on the local cache vector, see *MVS/ESA Programming: Sysplex Services Reference*.

System action: DB2 downgrades the expansion request to the size that can be supported by the available local cache vector storage (*size2*).

System programmer response: Free local cache vector storage by contracting or deleting other buffer pools that are connected to a group buffer pool. Then reattempt the pool expansion by reissuing the ALTER BUFFERPOOL command.

DSNB541I *csect-name* THE TOTAL ACTIVE PGFIX YES BUFFER POOL STORAGE EXCEEDS THE DB2 ALLOWED REAL STORAGE CAPACITY WHEN ALLOCATING STORAGE FOR BUFFER POOL *bpname* REAL STORAGE CAPACITY = *rsc* MB.

Explanation: When allocating storage for a PGFIX YES buffer pool, DB2 detects that the total storage allocated for the PGFIX YES buffer pools will exceed 80% of the real storage capacity on this z/OS image. The buffer pool name identified on this message (*bpname*) will be allocated with the PGFIX NO attribute. The real storage capacity of this z/OS image (*rsc*) is displayed in this message in units of megabytes (1 MB = 2**20 bytes).

System action: The buffer pool is created using the PGFIX NO attribute.

System programmer response: Either allocate more real storage to this z/OS image, reduce the requested size of the buffer pool or use the PGFIX NO attribute.

DSNB542I *csect-name* **THE TOTAL ACTIVE PGFIX YES BUFFER POOL STORAGE EXCEEDS THE DB2 ALLOWED REAL STORAGE CAPACITY WHEN EXPANDING STORAGE FOR BUFFER POOL** *bpname* **CURRENT VPSIZE =** *cbpsize* **NEW VPSIZE =** *nbpsize* **REAL STORAGE CAPACITY =** *rsc* **MB.**

Explanation: When expanding the VPSIZE for an active PGFIX YES buffer pool, DB2 detects that the total storage allocated for all active PGFIX YES buffer pools will exceed 80% of the real storage capacity on this z/OS image. The ALTER BUFFERPOOL command fails and the buffer pool will continue to be active with the current VPSIZE value.

The buffer pool attributes and real storage capacity are:

bpname
The buffer pool name.

cbpsize
The current VPSIZE for the buffer pool.

nbpsize
The new VPSIZE that was specified on the ALTER BUFFERPOOL command.

rsc The real storage capacity of this z/OS image in units of megabytes (1 MB = 2**20 bytes).

System action: The ALTER BUFFERPOOL command fails. The buffer pool continues to be active with the current VPSIZE.

System programmer response: Either allocate more real storage to this z/OS image, reduce the requested size of the buffer pool or use the PGFIX NO attribute.

DSNB543I **THE PGFIX ATTRIBUTE IS ALTERED FOR BUFFER POOL** *bpname* **CURRENT ATTRIBUTE =** *cpgfix* **NEW ATTRIBUTE =** *npgfix* **THE NEW ATTRIBUTE IS IN PENDING STATE.**

Explanation: This message is issued in response to an ALTER BUFFERPOOL command to indicate that the requested change to the PGFIX attribute has been accepted. The new PGFIX value will take effect the next time the buffer pool is allocated.

bpname
The buffer pool name.

cpgfix The current PGFIX value for the buffer pool.

npgfix The new PGFIX value that was specified on the ALTER BUFFERPOOL command.

System action: Processing continues.

DSNB551I **BSDS READ ERROR, RC=** *integer*, **RSN=***integer*

Explanation: This message is issued in response to a failure of a READ function on the bootstrap data set (BSDS). Users should check any previous error messages issued by MVS and DB2 against the corresponding BSDS. RC and RSN indicate an error return code and reason code from the BSDS function.

System action: Processing continues.

Operator response: Notify the system programmer.

System programmer response: Analyze the return and reason codes to determine what type of error occurred. Refer to message DSNJ107I for more information.

DSNB552I **BSDS WRITE ERROR, RC=** *integer*, **RSN=** *integer*

Explanation: This message is issued in response to a failure of a WRITE function on the bootstrap data set (BSDS). Check any previous error messages issued by MVS and DB2 against the corresponding BSDS. RC and RSN indicate an error return code and reason code from the BSDS function.

System action: Processing continues.

Operator response: Notify the system programmer.

System programmer response: Analyze the return and reason codes to determine what type of error occurred. Refer to message DSNJ108I for more information.

DSNB553I **BSDS INSERT ERROR, RC=** *integer*, **RSN=** *integer*

Explanation: This message is issued in response to a failure of an INSERT function on the bootstrap data set (BSDS). Check any previous error messages issued by MVS and DB2 against the corresponding BSDS. RC and RSN indicate an error return code and reason code from the BSDS function.

System action: Processing continues.

Operator response: Notify the system programmer.

System programmer response: Analyze the return and reason codes to determine what type of error occurred. For more information, refer to message DSNJ109I or other DSNJ messages related to BSDS.

DSNB601I **BUFFER POOL** *bpname* **FULL**

Explanation: A buffer is required to access a database page that is not currently contained within the buffer pool. A buffer cannot be assigned because all buffers in the buffer pool are in a nondisplaceable state. (Buffers are in a nondisplaceable state when they contain actively referenced pages or updated pages for which writes are pending.)

This message can occur if there is an overload of concurrent activity for the buffer pool size. The variable *bpname* is the buffer pool external name.

System action: Reason code 00C200E9 and the buffer pool external name are recorded in the cursor table (CT), which is available in the SQL communication area (SQLCA) or messages. An SQLCODE -904 (resource not available) is returned to the user.

Operator response: Notify the system programmer.

System programmer response: Do one of the following:
- Reduce concurrent activity by reducing the number of DB2 threads.
- Use the ALTER BUFFERPOOL command to increase the size of the buffer pool.

DSNB602I UNABLE TO CREATE BUFFER POOL *bpname*. BUFFER POOL SIZE IS ZERO

Explanation: An attempt to create the buffer pool failed while opening a DB2 table space or index that specified its use. The buffer pool size for *bpname* was zero. This indicates that the use of the pool is invalid.

System action: Reason code 00C200E0 and the buffer pool external name are recorded in the cursor table (CT), which is available in the SQL communication area (SQLCA) or messages. An SQLCODE -904 (resource not available) is returned to the user.

Operator response: Notify the system programmer.

System programmer response: Use the ALTER TABLESPACE or ALTER INDEX statement to modify affected table spaces or indexes to indicate a usable buffer pool. Use the DISPLAY BUFFERPOOL command to determine a usable buffer pool. If necessary, use the ALTER BUFFERPOOL command to activate the failing buffer pool by setting its VPSIZE to a nonzero value.

DSNB603I INSUFFICIENT STORAGE TO CREATE/EXPAND BUFFER POOL *bpname* TO SPECIFIED SIZE *integer*. BUFFER POOL SIZE IS NOW *integer*.

Explanation: A buffer pool could not be created or expanded to the requested size. Insufficient storage exists within the *ssnm*DBM1 address space to build the buffer pool. The buffer pool size is set by the ALTER BUFFERPOOL command and can be queried by using the DISPLAY BUFFERPOOL command.

System action: The buffer pool is created or expanded to the size allowed by the available storage. If the buffer pool cannot be created at all, and the creation request was triggered by a page set open by an application, then an SQLCODE -904 (resource not available) is returned to the user. Reason code 00C200E8 and the buffer pool external name are recorded in the cursor table (CT), which is available in the SQL communication area (SQLCA) or messages.

Operator response: Notify the system programmer.

System programmer response: Analyze the storage content of the *ssnm*DBM1 address space to determine further action. If appropriate, use the ALTER TABLESPACE or ALTER INDEX statement to use a different buffer pool. Otherwise, use ALTER BUFFERPOOL commands to balance storage use for all buffer pools.

DSNB605I UNABLE TO CREATE BUFFER POOL *bpname* TO SPECIFIED SIZE DURING STARTUP/RESTART. INSUFFICIENT VIRTUAL STORAGE

Explanation: An attempt to create the buffer pool failed at the time that DB2 was performing its startup or restart processing. Insufficient virtual storage exists within the *ssnm*DBM1 address space for building the buffer pool. The name of the buffer pool is *bpname*.

System action: The buffer pool is not created. The table space or index open request, which triggered the attempt to create the buffer pool, fails. An SQLCODE -904 (resource not available) is returned to the user. Reason code 00C200E8 and the buffer pool external name are recorded in the cursor table (CT), which is available in the SQL communication area (SQLCA) or messages.

Operator response: Notify the system programmer. This message remains on the console until the operator deletes it.

System programmer response: Analyze the virtual storage content of the DSNDBM1 *ssnm*DBM1 address space to determine further action. If appropriate, use ALTER BUFFERPOOL commands to balance virtual storage use for all buffer pools.

DSNB609I BUFFER POOL *bpname* CREATED USING DEFAULT SIZE *integer* DURING STARTUP/RESTART. BUFFER POOL SIZE IS ZERO

Explanation: While activating a buffer pool during DB2 startup or restart processing, the buffer pool size was zero. A zero value might be caused by an unusable bootstrap data set (BSDS) or by a DB2 abnormal termination after buffer pool size was altered to zero. The name of the buffer pool is *bpname* and the default buffer pool size is *integer*.

System action: DB2 creates the buffer pool using the startup restart default value. For 4KB/32KB page-type buffer pools, the default values are 1000 and 12, respectively.

Operator response: Notify the system programmer. This message remains on the console until deleted by the operator.

System programmer response: Use the ALTER TABLESPACE or ALTER INDEX statement to modify

the affected table spaces or indexes to indicate a usable buffer pool. To determine a usable buffer pool, use the DISPLAY BUFFERPOOL command. If necessary, use the ALTER BUFFERPOOL command to activate the failing buffer pool by setting the VPSIZE to a nonzero value.

DSNB610I *csect-name* **MAXIMUM ALLOCATABLE BUFFER POOL STORAGE OF** *int1***MB HAS BEEN REACHED. UNABLE TO CREATE/EXPAND BUFFER POOL** *bp-name* **TO SPECIFIED SIZE** *int2***. BUFFER POOL IS NOW** *int3*

Explanation: A buffer pool could not be created or expanded to the requested size. The maximum amount of allowable buffer pool storage, indicated by *int1*, has been reached. The maximum allowable buffer pool storage is calculated to be twice the amount of real storage that is available on the z/OS image, or 1TB, whichever is less.

bpname
Indicates the name of the buffer pool that was affected.

int2 Indicates the size of the buffer pool that was requested.

int3 Indicates the actual size of the buffer pool that was created/expanded.

System action: The buffer pool is created or expanded to the size allowed by the available storage within the maximum. If the maximum has already been reached, then the buffer pool is allocated at a minimum size as follows:

- 2,000 pages for 4K page size
- 1,000 pages for 8K page size
- 500 pages for 16K page size
- 250 pages for 32K page size

Once the maximum has been reached, buffer pool sizes are not allowed to be increased.

System programmer response: Analyze the buffer pool sizes and determine if any can be decreased, or consider adding more memory.

DSNB740I *csect-name* **ATTEMPTING TO ESTABLISH DUPLEXING FOR GROUP BUFFER POOL** *gbpname* **REASON =** *reason*

Explanation: DB2 has been requested to establish duplexing for the named group buffer pool. *reason* indicates the reason why duplexing was initiated. *reason* can be one of the following:

OPERATOR
The operator initiated a SETXCF command to initiate duplexing.

POLICY
The currently active CFRM policy indicated that duplexing should be automatically established for the group buffer pool.

This message is issued by each DB2 member participating in establishing duplexing.

System action: The DB2 members that are actively connected to the named group buffer pool participate in establishing duplexing for the group buffer pool. Access to the group buffer pool is quiesced while duplexing is being established.

DSNB741I *csect-name* **DUPLEXING COULD NOT BE ESTABLISHED FOR GROUP BUFFER POOL** *gbpname* **REASON =** *reason*

Explanation: DB2 was unable to establish duplexing for the named group buffer pool. *reason* indicates the DB2 reason code indicating why duplexing could not be established.

System action: The group buffer pool goes back to simplex mode.

System programmer response: See the explanation for the reason code given.

DSNB742I *csect-name* **DUPLEXING HAS BEEN SUCCESSFULLY ESTABLISHED FOR GROUP BUFFER POOL** *gbpname*

Explanation: The process to establish duplexing for the named group buffer pool has successfully completed.

This message is preceded by message DSNB740I which indicates the reason why duplexing was initiated. This message is issued by each DB2 member participating in establishing duplexing.

System action: Processing of mainline requests is resumed for the group buffer pool. Writes of changed pages to the group buffer pool will be duplexed writes from this point forward.

DSNB743I *csect-name* **DUPLEXING IS BEING STOPPED FOR GROUP BUFFER POOL** *gbpname* **FALLING BACK TO PRIMARY REASON =** *reason1* **DB2 REASON CODE =** *reason2*

Explanation: Duplexing is being stopped for the named group buffer pool. To stop the duplexing, the secondary group buffer pool will be deallocated, and DB2 will go back to simplex mode on the primary group buffer pool.

reason1 indicates why duplexing was stopped:

OPERATOR
The operator initiated an MVS command to stop duplexing.

POLICY
A new CFRM policy has been activated which specifies DUPLEX(DISABLED) for the group buffer pool.

LOSSCONNSEC
One or more of the DB2 members lost connectivity to the secondary group buffer pool structure.

STRFAILSEC
The secondary group buffer pool structure failed.

DB2REASON
DB2 decided to stop duplexing. The reason code given in *reason2* indicates why DB2 stopped duplexing.

DUPCONNCHG
Duplexing rebuild stopped because of insufficient connectivity due to a change in the set of connectors.

If *reason1* indicates 'DB2REASON', then *reason2* specifies a DB2 reason code that indicates why DB2 stopped duplexing. If *reason1* is not 'DB2REASON', then *reason2* is zero.

This message is issued by each DB2 member connected to the duplexed group buffer pool.

System action: The secondary group buffer pool structure is discarded, and DB2 enters back into simplex mode on the primary group buffer pool structure. If DUPLEX(AUTO) is specified in the active CFRM policy for the group buffer pool, then duplexing will be automatically reinitiated, if duplexing is still possible.

System programmer response: If the rebuild stopped because of "DB2REASON", see the explanation for the reason code given. If duplexing is not automatically reinitiated and duplexing is desired for the group buffer pool, then use the MVS SETXCF command to reestablish duplexing.

DSNB744I *csect-name* **DUPLEXING IS BEING STOPPED FOR GROUP BUFFER POOL** *gbpname*, **SWITCHING TO SECONDARY REASON =** *reason1* **DB2 REASON CODE =** *reason2*

Explanation: Duplexing is being stopped for the named group buffer pool. To stop the duplexing, the primary group buffer pool will be deallocated, and DB2 will switch to the the secondary group buffer pool to go back to simplex mode.

reason1 indicates why duplexing was stopped:

OPERATOR
The operator initiated an MVS command to stop duplexing.

POLICY
A new CFRM policy has been activated which specifies DUPLEX(DISABLED) for the group buffer pool.

LOSSCONNPRI
One or more of the DB2 members lost connectivity to the primary group buffer pool structure.

STRFAILPRI
The primary group buffer pool structure failed.

DB2REASON
DB2 decided to stop duplexing. The reason code given in *reason2* indicates why DB2 stopped duplexing.

If *reason1* indicates 'DB2REASON', then *reason2* specifies a DB2 reason code that indicates why DB2 stopped duplexing. If *reason1* is not 'DB2REASON', then *reason2* is zero.

This message is issued by each DB2 member connected to the duplexed group buffer pool.

System action: The primary group buffer pool structure is discarded, and DB2 enters back into simplex mode on the secondary group buffer pool structure. If DUPLEX(ENABLED) is specified in the active CFRM policy for the group buffer pool, then duplexing will be automatically reinitiated, if duplexing is still possible.

System programmer response: If the rebuild stopped because of "DB2REASON", see the explanation for the reason code given. If duplexing is not automatically reinitiated and duplexing is desired for the group buffer pool, then use the MVS SETXCF command to reestablish duplexing.

DSNB745I *csect-name* **THE TRANSITION BACK TO SIMPLEX MODE HAS COMPLETED FOR GROUP BUFFER POOL** *gbpname*

Explanation: The named group buffer pool has completed the transition from duplex mode back to simplex mode.

This message is preceded either by message DSNB743I or by message DSNB744I which indicates the reason why duplexing was stopped. This message is issued by each DB2 member connected to the group buffer pool.

System action: The group buffer pool is now back in simplex mode which means that writes of changed pages to the group buffer pool are now simplexed. If DUPLEX(ENABLED) is specified in the active CFRM

policy for the group buffer pool, then duplexing will be automatically reinitiated, if duplexing is still possible. This would mean that the group buffer pool is only temporarily back into simplex mode (it is soon to be entered back into duplex mode).

DSNB749I GBPCACHE NO IS NOT ALLOWED FOR GROUP BUFFER POOL *gbpname* **BECAUSE IT IS DUPLEXED**

Explanation: This message is issued in response to an ALTER GROUPBUFFERPOOL command that requests a change of the GBPCACHE attribute from YES to NO, but because the group buffer pool is duplexed, the change from YES to NO is not allowed.

System action: The request to change from GBPCACHE(YES) to GBPCACHE(NO) is rejected.

System programmer response: GBPCACHE(NO) and duplexing are mutually exclusive. If you want GBPCACHE(NO), then you must put the group buffer in simplex mode before specifying GBPCACHE(NO). To put a duplexed group buffer pool in simplex mode, do any of the following procedures:

- Activate a new CFRM policy that specifies DUPLEX(DISABLED) for the group buffer pool. This is the preferred method if you do not intend to duplex the group buffer pool again for awhile.
- Ensure that DUPLEX(AUTO) is not specified in the active CFRM policy for the group buffer pool, and then issue the MVS command SETXCF STOP,REBUILD,DUPLEX.

DSNB750I DISPLAY FOR GROUP BUFFER POOL *gbpname* **FOLLOWS**

Explanation: This message is issued in response to a a DISPLAY GROUPBUFFERPOOL command. It signals the beginning of a display for group buffer pool *gbpname* if the following are true:

- It meets the selection criteria on the command.
- The MVS CFRM administrative policy information for the group buffer pool is accessible.

If the CFRM policy information is unavailable, message DSNB751I signals the partial display of group buffer pool information.

The series of messages DSNB755I through DSNB759I provide information on the group buffer pool's allocation characteristics. Messages DSNB765I through DSNB769I provide information on DB2 connections to the group buffer pool, while DSNB770I through DSNB788I provide statistics on group buffer pool usage. Note that all message numbers in the range might not be used. Message DSNB790I ends the display.

DSNB751I MVS CFRM POLICY INFORMATION NOT AVAILABLE FOR GROUP BUFFER POOL *gbpname* **MVS IXCQUERY RETURN CODE =** *retcode* **MVS IXCQUERY REASON CODE =** *reason* **A PARTIAL DISPLAY FOLLOWS**

Explanation: DB2 could not retrieve group buffer pool policy information in response to a DISPLAY GROUPBUFFERPOOL command. Either the MVS query request (IXCQUERY) did not successfully complete, or the backing structure for the group buffer pool is not defined in the MVS CFRM administrative policy. To determine the exact reason, examine return code *retcode* and reason code *reason* returned by MVS on the query request. For an explanation of the IXCQUERY codes, see *MVS/ESA Programming: Sysplex Services Reference*. If the IXCQUERY request did not complete successfully, no CONNLIST information, even if requested, is provided.

System action: Processing continues, but the information displayed is incomplete.

DSNB752I GROUP DETAIL STATISTICS ARE NOT AVAILABLE FOR THE GROUP BUFFER POOL BECAUSE A REBUILD IS IN PROGRESS

Explanation: Because a rebuild is in progress for the group buffer pool, DB2 cannot retrieve group buffer pool group (GDETAIL) statistics in response to a DISPLAY GROUPBUFFERPOOL command. A preceding DSNB750I or DSNB751I message identifies the associated group buffer pool.

System action: Processing continues.

DSNB753I GROUP BUFFER POOL STRUCTURE OWNER INFORMATION NOT AVAILABLE IRLM NOTIFY RETURN CODE = *retcode* **IRLM NOTIFY REASON CODE =** *reason*

Explanation: DB2 could not retrieve group buffer pool structure owner information for message DSNB798I because the IRLM Notify request did not successfully complete. A preceding DSNB750I or DSNB751I message identifies the associated group buffer pool.

retcode Return code from IRLM on the Notify request

reason Reason code from IRLM on the Notify request

For an explanation of IRLM return and reason codes, see *DB2 Codes*.

System action: Processing continues.

DSNB755I **DB2 GROUP BUFFER POOL STATUS CONNECTED =** *status1* **CURRENT DIRECTORY TO DATA RATIO =** *ratio1* **PENDING DIRECTORY TO DATA RATIO =** *ratio2* **CURRENT GBPCACHE ATTRIBUTE =** *gcache1* **PENDING GBPCACHE ATTRIBUTE =** *gcache2*

Explanation: In response to a DISPLAY GROUPBUFFERPOOL command, this is the first of two messages displaying information about the status of the group buffer pool as it is known to DB2. This message is followed by DSNB756I. A preceding DSNB750I or DSNB751I message identifies the associated group buffer pool.

status1 Whether this DB2 is connected to the group buffer pool:

YES This DB2 is actively connected to the group buffer pool.

NO This DB2 is not actively connected to the group buffer pool.

ratio1 The ratio of directory entries to data pages, as specified for the current allocation of this group buffer pool.

ratio2 The desired ratio of directory entries to data pages to take effect at the next reallocation of the group buffer pool.

gcache1 The GBPCACHE attribute as specified for the current allocation of this group buffer pool.

gcache2 The value for GBPCACHE attribute that will take effect at the next reallocation of the group buffer pool.

DSNB756I **CLASS CASTOUT THRESHOLD =** *threshold1* **% GROUP BUFFER POOL CASTOUT THRESHOLD =** *threshold2* **% GROUP BUFFER POOL CHECKPOINT INTERVAL =** *interval* **MINUTES RECOVERY STATUS =** *status1* **AUTOMATIC RECOVERY =** *auto-rec-parm*

Explanation: This is the second of two messages displaying information about the status of the group buffer pool as it is known to DB2. If DB2 was successful in getting MVS CFRM policy information for the group buffer pool, messages DSNB757I through DSNB759I immediately follow this message. A preceding DSNB750I or DSNB751I message identifies the associated group buffer pool.

threshold1
The threshold at which class castouts are to be initiated. It is expressed as a percentage of the group buffer pool size.

threshold2
The threshold at which castout is to be initiated at the group buffer pool level. It is expressed as a percentage of the group buffer pool size.

interval The time, in minutes, between successive initiations of group buffer pool checkpoint.

status1 The recovery status for the group buffer pool:

DAP Damage assessment pending. DB2 is in the process of doing damage assessment for the group buffer pool in response to a coupling facility structure failure. The group buffer pool is temporarily stopped internally by DB2 and is unavailable for use until the damage assessment process completes.

NORMAL
There is no pending recovery action for this group buffer pool.

auto-rec-parm
Automatic recovery parameter specification:

YES DB2 should automatically initiate recovery for the objects using this group buffer pool following a failure of the group buffer pool. This is the default.

NO DB2 should not initiate automatic group buffer pool recovery. If the group buffer pool fails, you must manually issue DB2 -START DATABASE commands to do the recovery.

DSNB757I **MVS CFRM POLICY STATUS FOR** *structure-name* **=** *status2* **MAX SIZE INDICATED IN POLICY =** *psize* **KB DUPLEX INDICATOR IN POLICY =** *dupstat* **CURRENT DUPLEXING MODE =** *dupmode* **ALLOCATED =** *status1*

Explanation: For a DISPLAY GROUPBUFFERPOOL command, this message gives the status for the structure associated with the group buffer pool. It is displayed only if DB2 was successful in querying the MVS CFRM administrative policy. It is immediately preceded by message DSNB756I, and if the structure is allocated, followed by messages DSNB758I and DSNB759I. A preceding DSNB750I message identifies the associated group buffer pool.

structure-name
The coupling facility structure name for this group buffer pool, such as DSNCAT_GBP0.

status2 MVS global status of the backing coupling facility structure:

NORMAL
Structure is defined in the current CFRM policy and has no current MVS restrictions.

CHNGP
Change pending in the structure policy.

dupstat The duplexing attribute for the coupling facility structure, as indicated by the DUPLEX parameter in the CFRM policy definition.

DISABLED
DUPLEX(DISABLED) is specified for the coupling facility structure. The DUPLEX parameter is not specified.

ENABLED
DUPLEX(ENABLED) is specified for the coupling facility structure.

ALLOWED
DUPLEX(ALLOWED) is specified for the coupling facility structure.

dupmode
Indication of current status of the group buffer pool in relation to duplexing.

SIMPLEX
Group buffer pool is currently in simplex mode.

DUPLEX
Group buffer pool is currently in duplex mode.

INTRAN
Group buffer pool is currently in a transition state. Either from simplex mode to duplex mode, or from duplex mode to simplex mode. The rebuild status indicator in message DSNB758I gives information about the transition state.

psize The size (in kilobytes) of the coupling facility structure as indicated in the CFRM policy definition.

status1 Whether the backing coupling facility structure is currently allocated:

YES The coupling facility structure is allocated in a coupling facility.

NO The coupling facility structure is not allocated in any coupling facility.

DSNB758I **ALLOCATED SIZE =** *asize* **KB VOLATILITY STATUS =** *volstat* **REBUILD STATUS =** *rbldstat* **CFNAME =** *cfname* **CFLEVEL - OPERATIONAL =** *cflevel1* **CFLEVEL - ACTUAL =** *cflevel2*

Explanation: This message is the first of two messages giving the allocation details for the group buffer pool on a DISPLAY GROUPBUFFERPOOL command. It is issued only if DB2 determines, by successfully querying the MVS CFRM policy, that the group buffer pool is allocated. It is immediately preceded by message DSNB757I. A preceding DSNB750I message identifies the associated group buffer pool.

asize The actual allocated group bufferpool structure size (in kilobytes).

volstat The volatility status of the backing group buffer pool structure for this group buffer pool:

VOLATILE
The backing coupling facility structure is volatile.

NON-VOLATILE
The backing coupling facility structure is non-volatile.

rbldstat Inidication of whether or not rebuild is in progress for the group buffer pool structure, and if so, which part of the rebuild process is happening:

NONE Rebuild is not in progress.

QUIESCE
Rebuild is in progress and the rebuild is currently in the rebuild quiesce phase.

CONNECT
Rebuild is in progress and the rebuild is currently in the rebuild connect phase.

CLEANUP
The rebuild has completed, and the connected DB2 members are making final preparations to use the new group buffer pool structure. If the duplexing status in message DSNB757I indicates 'INTRAN', then the 'CLEANUP' indication means that DB2 is in the final stages of transitioning from duplex mode back to simplex mode, switching to the secondary group buffer pool.

STOPPING
The rebuild is stopping, and the connected DB2 members are making preparations to fall back to the original group buffer pool structure. If the duplexing status in message DSNB757I indicates 'INTRAN', then the 'STOPPING' indication means that DB2 is transitioning from duplex mode back to simplex mode, reverting back to the primary group buffer pool.

DUPLEXED
The group buffer pool is in steady-state duplexing mode.

SWITCHING
DB2 is in the initial phase of transitioning from duplex mode back to simplex mode, switching to the secondary group buffer pool. The group buffer pool is transitioning from duplex mode back to simplex mode, switching to the secondary group buffer pool.

cfname The name of the coupling facility in which the group buffer pool is allocated. If the group buffer pool is duplexed, this is the coupling facility name associated with the primary group buffer pool.

cflevel1 The coupling facility level from a DB2 functionality perspective for the coupling facility in which this group buffer pool is allocated. If the group buffer pool is duplexed, *cflevel1* is the coupling facility level asscociated with the primary group buffer pool.

cflevel2 The coupling facility level for the coupling facility in which this group buffer pool is allocated. If the group buffer pool is duplexed, *cflevel2* is the coupling facility level asscociated with the primary group buffer pool."

DSNB759I NUMBER OF DIRECTORY ENTRIES = *dir#* NUMBER OF DATA PAGES = *data#* NUMBER OF CONNECTIONS = *conn#*

Explanation: This message is the second of two messages giving the allocation details for the group buffer pool for a DISPLAY GROUPBUFFERPOOL command. It is issued only if DB2 determines, by successfully querying the MVS CFRM administrative policy, that the group buffer pool is allocated. It is immediately preceded by message DSNB758I. A preceding DSNB750I message identifies the associated group buffer pool.

dir# Actual number of directory entries allocated for the coupling facility structure.

data# Actual number of data pages allocated for the coupling facility structure.

conn# The number of DB2 subsystems that are connected to the coupling facility structure. This includes both active connections and failed-persistent connections. The CONNLIST option of the DISPLAY GROUPBUFFERPOOL command gives details of the connection status.

DSNB760I GBPCACHE OPTION SET TO *value* FOR *gbpname*. IT WILL TAKE EFFECT ON THE NEXT REALLOCATION

Explanation: This message is issued in response to an ALTER GROUPBUFFERPOOL command that requests a change in the GBPCACHE attribute. It is also issued for the first ALTER GROUPBUFFERPOOL command for a group buffer pool, even when the GBPCACHE option is not specified on the command. (In this case, the message shows YES for the GBPCACHE option.) The new setting takes effect the next time the coupling facility cache structure is allocated.

System programmer response: Use the MVS command SETXCF START, REBUILD command when you are ready for the change to take effect.

System action: Processing continues.

DSNB761I THE NEW SETTING WILL TAKE EFFECT AFTER THE GBPCACHE ATTRIBUTE HAS BEEN CHANGED TO YES

Explanation: This message is issued in response to an ALTER GROUPBUFFERPOOL command that requests a change to RATIO, CLASST, GBPOOLT or GBPCHKPT when GBPCACHE NO is also specified on the command or when the current GBPCACHE attribute is NO. The new setting takes effect the next time the coupling facility cache structure is allocated after the GBPCACHE attribute has been changed to YES.

System programmer response: Use the ALTER GROUPBUFFERPOOL command to change the GBPCACHE attribute to YES and the MVS command SETXCF START, REBUILD when you are ready for the change to take effect.

System action: Processing continues.

DSNB762I DUPLEXING STATISTICS FOR *secondary-gbpname* WRITES FAILED DUE TO LACK OF STORAGE = *s2* CHANGED

Explanation: This message is displayed as a result of the DISPLAY GROUPBUFFERPOOL command with the GDETAIL option if the group buffer pool is duplexed. It lists the statistics on writes to the secondary group buffer pool and the total number of changed pages for the secondary group buffer pool. *secondary-gbpname* is the normal group buffer pool name (such as GBP0) suffixed with '-SEC'.

s2 **Write miss cache full** counter. Number of times a coupling facility write request was not completed because of a lack of storage in the secondary group buffer pool.

For information about the Group Detail Report, see *DB2 Data Sharing: Planning and Administration* .

DSNB763I *gbpname* **IS NOT DEFINED WITH GBPCACHE NO**

Explanation: This message is issued in response to a DISPLAY GROUPBUFFERPOOL command with TYPE(NOCACHE) and either one name or a list of names specified in the selection list. The specified group buffer pool is not defined with GBPCACHE NO, and therefore does not meet the qualification to be displayed. This message is issued once for each group buffer pool that does not match the "GBPCACHE NO" criterion.

System action: The display is terminated for this group buffer pool. Command processing continues, if there are other group buffer pool names to process.

DSNB764I **DUPLEXING STATISTICS FOR** *secondary-gbpname* **WRITES FAILED DUE TO LACK OF STORAGE =** *s2* **ASYNCHRONOUS COMPLETION CHECKS =** *s3*

Explanation: This message is displayed as a result of the DISPLAY GROUPBUFFERPOOL command with the MDETAIL option if the group buffer pool is duplexed. It lists the statistics on writes to the secondary group buffer pool. *secondary-gbpname* is the normal group buffer pool name (e.g. GBP0) suffixed with '-SEC'.

s2 Coupling facility write requests that were not completed because of a lack of storage in the secondary group buffer pool.

s3 The number of times DB2 checked for the completion of the write of a changed page to the secondary group buffer pool, but the write had not yet completed and thus the transaction had to wait until the write to the secondary group buffer pool completed.

For information about the Member Detail Report, see *DB2 Data Sharing: Planning and Administration*.

DSNB765I **NO CONNLIST REPORT IS GENERATED. THERE ARE NO CONNECTIONS TO THIS GROUP BUFFER POOL.**

Explanation: DB2 displays this message in response to a DISPLAY GROUPBUFFERPOOL command with the CONNLIST option when there are currently no connection names to report for the group buffer pool. A preceding DSNB750I message identifies the associated group buffer pool.

DSNB766I **THE CONNLIST REPORT FOLLOWS**

Explanation: DB2 displays this message in response to a DISPLAY GROUPBUFFERPOOL command with the CONNLIST option to indicate the beginning of the CONNLIST display for this group buffer pool. A preceding DSNB750I message identifies the associated group buffer pool.

DSNB767I **CONNECTION NAME =** *conname*, **CONNECTION STATUS =** *status* **CONNECTOR'S RELEASE =** *DB2 release*

Explanation: This message contains the display information that results from a DISPLAY GROUPBUFFERPOOL command with the CONLIST option. One DSNB767I message is issued for each defined connection to the group buffer pool. The associated group buffer pool is identified on a preceding DSNB750I message.

conname
Name of the connection. DB2 subsystems use the DB2 member name as the connection name.

status Status of the connection.

A Active.

F Failed-Persistent.

D Dually connected to both the original structure and the new structure during a rebuild.

DB2 release
The release of the connected DB2.

DSNB768I **THERE ARE NO GROUP BUFFER POOLS DEFINED WITH GBPCACHE NO**

Explanation: This message is issued in response to a DISPLAY GROUPBUFFERPOOL command with the TYPE(NOCACHE) option and an asterisk (*) specified as the selection list when DB2 determines that there are no group buffer pools defined with the GBPCACHE(NO) attribute.

System action: Command processing is terminated.

DSNB769I **THE CONNLIST REPORT IS COMPLETE**

Explanation: DB2 displays this message in response to a DISPLAY GROUPBUFFERPOOL command with the CONNLIST option to indicate the end of the CONNLIST display for this group buffer pool. A preceding DSNB750I message identifies the associated group buffer pool.

DSNB770I **MEMBER DETAIL STATISTICS ARE NOT AVAILABLE FOR THE GROUP BUFFER POOL. THIS DB2 HAS NEVER BEEN CONNECTED TO THE GROUP BUFFER POOL.**

Explanation: This message is issued in response to a DISPLAY GROUPBUFFERPOOL command with

MDETAIL specified. MDETAIL statistics are not available for the requested group buffer pool because this DB2 has not connected to the group buffer pool since startup. A preceding DSNB750I or DSNB751I message identifies the associated group buffer pool.

System action: Processing continues.

DSNB771I INCREMENTAL MEMBER DETAIL STATISTICS SINCE *basetime*

Explanation: This message is displayed in response to the DISPLAY GROUPBUFFERPOOL command with MDETAIL(INTERVAL) specified. It indicates the start time of the interval for which the statistics were accumulated. This is the time of the previous MDETAIL incremental display on this DB2 or the time DB2 first connected to the group buffer pool if there was no previous MDETAIL incremental display. A preceding DSNB750I or DSNB751I message identifies the associated group buffer pool.

System action: Processing continues.

DSNB772I CUMULATIVE MEMBER DETAIL STATISTICS SINCE *basetime*

Explanation: This message is displayed in response to the DISPLAY GROUPBUFFERPOOL command with MDETAIL(*) specified. It indicates the start time of the interval for which the statistics were accumulated. This is the time this DB2 first connected to the group buffer pool. A preceding DSNB750I or DSNB751I message identifies the associated group buffer pool.

System action: Processing continues.

DSNB773I MEMBER DETAIL STATISTICS SYNCHRONOUS READS DUE TO BUFFER INVALIDATION DATA RETURNED = *r1* **DATA NOT RETURNED =** *r2*

Explanation: This is the first of eight messages containing local statistics, displayed as a result of the DISPLAY GROUPBUFFERPOOL command with the MDETAIL option. It lists some synchronous read statistics for the group buffer pool. A preceding DSNB750I or DSNB751I message identifies the associated group buffer pool.

The counter descriptions are listed below. In the descriptions, the term “synchronous” means that the coupling facility interaction is done under the DB2 user’s execution unit.

r1 Synchronous coupling facility read requests issued because the virtual buffer was marked “invalid”. Data is successfully returned from the group buffer pool.

r2 Synchronous coupling facility read requests issued because the virtual buffer was marked “invalid”. Data is *not* returned from the group buffer pool.

DSNB774I DUE TO DATA PAGE NOT IN BUFFER POOL DATA RETURNED = *r3* **DATA NOT RETURNED =** *r4*

Explanation: This is the second of eight messages containing local statistics, displayed as a result of the DISPLAY GROUPBUFFERPOOL command with the MDETAIL option. It lists more local synchronous read statistics for the group buffer pool. A preceding DSNB750I or DSNB751I message identifies the associated group buffer pool.

The counter descriptions are listed below. In the descriptions, the term ″synchronous″ means that the coupling facility interaction is done under the DB2 user’s execution unit.

r3 Synchronous coupling facility read requests issued because the required data was not in the buffer pool buffer. Data is successfully returned from the group buffer pool.

r4 Synchronous coupling facility read requests issued because the required data was not in the buffer pool buffer. Data is *not* returned from the group buffer pool.

DSNB776I SYNCHRONOUS WRITES CHANGED PAGES = *w1* **CLEAN PAGES =** *w2*

Explanation: This is the fifth of eight messages containing local statistics, displayed as a result of the DISPLAY GROUPBUFFERPOOL command with the MDETAIL option. It lists some local write statistics for the group buffer pool. A preceding DSNB750I or DSNB751I message identifies the associated group buffer pool.

The counter descriptions are listed below. In the descriptions, the term ″synchronous″ means that the coupling facility interaction is done under the DB2 user’s execution unit.

w1 Number of changed pages that were written synchronously, under a user’s execution unit, to the coupling facility.

w2 Number of clean pages that were written synchronously, under a user’s execution unit, to the coupling facility

---

DSNB777I ASYNCHRONOUS WRITES CHANGED PAGES = *w3* **CLEAN PAGES =** *w4* **FAILED WRITES DUE TO LACK OF STORAGE =** *w5*

Explanation: This is the sixth of eight messages containing local statistics, displayed as a result of the DISPLAY GROUPBUFFERPOOL command with the MDETAIL option. It lists local write and

cross-invalidation statistics for the group buffer pool. A preceding DSNB750I or DSNB751I message identifies the associated group buffer pool.

The counter descriptions are listed below. In the descriptions, the term "asynchronous" means that the coupling facility interaction is done under a DB2 system execution unit, such as with a prefetch.

w3 Number of changed pages that were written asynchronously, under a system execution unit, to the coupling facility

w4 Number of clean pages that were written asynchronously, under a system execution unit, to the coupling facility.

w5 Coupling facility write requests that were not completed because of a lack of storage in the group buffer pool.

DSNB778I CASTOUT THRESHOLDS DETECTED FOR CLASSES = *t1* FOR GROUP BUFFER POOL = *t2* GBP CHECKPOINTS TRIGGERED = *k1* PARTICIPATION IN REBUILD = *k2*

Explanation: This is the seventh of eight messages containing local statistics, displayed as a result of the DISPLAY GROUPBUFFERPOOL command with the MDETAIL option. It lists local castout threshold statistics for the group buffer pool and also group buffer pool checkpoint and rebuild statistics. A preceding DSNB750I or DSNB751I message identifies the associated group buffer pool.

Counters:

t1 Number of times the class castout threshold was detected

t2 Number of times the group buffer pool castout threshold was detected

k1 Number of group buffer pool checkpoints that were triggered.

k2 Number of times that this member participated in a rebuild of the group buffer pool structure.

DSNB780I GROUP DETAIL STATISTICS ARE NOT AVAILABLE FOR THE GROUP BUFFER POOL. THIS DB2 IS NOT CURRENTLY CONNECTED TO THE GROUP BUFFER POOL.

Explanation: DB2 could not retrieve group buffer pool group (GDETAIL) statistics in response to a DISPLAY GROUPBUFFERPOOL command because this DB2 subsystem is not currently connected to the group buffer pool. A preceding DSNB750I or DSNB751I message identifies the associated group buffer pool.

System action: Processing continues.

DSNB781I GROUP DETAIL STATISTICS ARE NOT AVAILABLE FOR THE GROUP BUFFER POOL. MVS IXLCACHE RETURN CODE = *retcode* MVS IXLCACHE REASON CODE = *reason*

Explanation: DB2 could not retrieve group buffer pool group (GDETAIL) statistics in response to a DISPLAY GROUPBUFFERPOOL command because the statistics request (IXLCACHE macro) did not successfully complete. A preceding DSNB750I or DSNB751I message identifies the associated group buffer pool.

retcode Return code from MVS on the statistics request

reason Reason code from MVS on the statistics request

For an explanation of IXLCACHE return and reason codes, see *MVS/ESA Programming: Sysplex Services Reference*.

System action: Processing continues.

DSNB782I INCREMENTAL GROUP DETAIL STATISTICS SINCE *basetime*

Explanation: This message is displayed by the DISPLAY GROUPBUFFERPOOL command with GDETAIL(INTERVAL) specified. It indicates the start time of the interval for which the statistics were accumulated. This is the time of the previous GDETAIL incremental display from this DB2 or the time the group buffer pool was allocated or reallocated, if there was no previous GDETAIL incremental display.

System action: Processing continues.

DSNB783I CUMULATIVE GROUP DETAIL STATISTICS SINCE *basetime*

Explanation: This message is displayed in response to the DISPLAY GROUPBUFFERPOOL command with GDETAIL(*) specified. It indicates the start time of the interval for which the statistics were accumulated. This is the time the group buffer pool was first allocated or reallocated.

System action: Processing continues.

DSNB784I GROUP DETAIL STATISTICS READS DATA RETURNED = *r1*

Explanation: This is the first of five messages containing group statistics, displayed as a result of the DISPLAY GROUPBUFFERPOOL command with the GDETAIL option. It lists the successful reads for the group buffer pool. A preceding DSNB750I or DSNB751I message identifies the associated group buffer pool. This message is always followed by messages DSNB785I through DSNB788I.

r1 "Read hit" counter. Number of times that a page was returned on a coupling facility read request.

For information about the Group Detail Report, see *DB2 Data Sharing: Planning and Administration* .

DSNB785I DATA NOT RETURNED DIRECTORY ENTRY EXISTED = *r2* DIRECTORY ENTRY CREATED = *r3* DIRECTORY ENTRY NOT CREATED = *r4*, *r5*

Explanation: This is the second of five messages containing group statistics, displayed as a result of the DISPLAY GROUPBUFFERPOOL command with the GDETAIL option. It continues the read statistics for the group buffer pool. The associated group buffer pool is identified in a preceding DSNB750I or DSNB751I message. This message is always preceded by message DSNB784I and followed by messages DSNB786I through DSNB788I.

All the "read counters" in this message represent cases where data was not returned on the coupling facility read request.

r2 "Read miss directory hit" counter. Number of times that a coupling facility read request specified a page for which a directory entry exists, but no data was cached for that page.

r3 "Read miss name assigned" counter. Number of times that a coupling facility read request specified a page for which a directory entry was created.

r4 "Read miss assignment suppressed" counter. Number of times that a coupling facility read request specified a page for which no directory entry exists and no directory entry is created.

r5 "Read miss cache full" counter. Number of times that a coupling facility read request specified a page for which no directory entry exists and no directory entry is created because of a lack of storage in the group buffer pool.

For information about the Group Detail Report, see *DB2 Data Sharing: Planning and Administration* .

DSNB786I WRITES CHANGED PAGES= *w1* CLEAN PAGES = *w2* FAILED DUE TO LACK OF STORAGE = *w3* CHANGED PAGES SNAPSHOT VALUE = *p1*

Explanation: This is the third of five messages containing group statistics, displayed as a result of the DISPLAY GROUPBUFFERPOOL command with the GDETAIL option. It is followed by messages DSNB787I and DSNB788I. It lists the statistics on writes to the group buffer pool and the total number of changed pages for the group buffer pool. The associated group buffer pool is identified in a preceding DSNB750I or DSNB751I message. This message is always preceded by messages DSNB784I and DSNB785I.

w1 "Changed page write hit" counter. Number of changed pages that were successfully written.

w2 "Clean page write hit" counter. NNumber of clean pages that were successfully written.

w3 "Write miss cache full" counter. Number of times a coupling facility write request was not completed because of a lack of storage in the group buffer pool.

p1 "Total changed" counter. Snapshot value of the current number of changed pages. The displayed value of this counter is not affected by whether this is an incremental or cumulative display.

For information about the Group Detail Report, see *DB2 Data Sharing: Planning and Administration* .

DSNB787I RECLAIMS FOR DIRECTORY ENTRIES = *d1* FOR DATA ENTRIES = *d2* CASTOUTS= *c1*

Explanation: This is the fourth of five messages containing group statistics, displayed as a result of the DISPLAY GROUPBUFFERPOOL command with the GDETAIL option. It is always preceded by messages DSNB784I through DSNB786I and followed by message DSNB788I. It lists the statistics on reclaims and castouts for the group buffer pool. The associated group buffer pool is identified in a preceding DSNB750I or DSNB751I message.

d1 "Castout" counter. Number of pages that were castout.

d2 "Data entry reclaim" counter. Number of times a name assignment required that a data page be reclaimed by the coupling facility.

c1 "Castout" counter. Number of castout operations performed.

For information about the Group Detail Report, see *DB2 Data Sharing: Planning and Administration* .

DSNB788I CROSS INVALIDATIONS DUE TO DIRECTORY RECLAIMS = *x1* DUE TO WRITES = *x2* EXPLICIT = *x3*

Explanation: This is the last of five messages containing group statistics, displayed as a result of the DISPLAY GROUPBUFFERPOOL command with the GDETAIL option. It lists the cross-invalidation (XI) statistics for the group buffer pool. The associated group buffer pool is identified in a preceding DSNB750I or DSNB751I message. This message is always preceded by messages DSNB784I through DSNB787I.

x1 "XI for directory reclaim" counter. Number of times a directory entry was reclaimed (stolen) and XI signals had to be sent (that is, the stolen directory entry had registered DB2 interest).

x2 "XI for write" counter. Number of times a coupling facility write request caused XI to one or more registered DB2s.

x3 "XI for explicit invalidation" counter. Number of times an explicit coupling facility XI request caused XI to one or more registered DB2s.

For more information about these counters, see *DB2 Data Sharing: Planning and Administration*.

DSNB789I REGISTER PAGE LIST = *r7* PAGES RETRIEVED = *r8*

Explanation: This is the fourth of eight messages containing local statistics, displayed as a result of the DISPLAY GROUPBUFFERPOOL command with the MDETAIL option. It lists more local read statistics for the group buffer pool. A preceding DSNB750I or DSNB751I message identifies the associated group buffer pool.

The counter descriptions are listed below.

r7 The number of requests to the coupling facility to register interest in a list of pages. DB2 does this when prefetch (sequential or list) is used when the following conditions are met:
- The DB2 member is running on a particular service level of MVS 5.2 or above
- The GBP is allocated in a coupling facility with CFLEVEL=2 or above.

r8 Coupling facility read requests that were done for prefetch to retrieve a page in the group buffer pool as a result of feedback from the request to register a list of pages.

DSNB790I DISPLAY FOR GROUP BUFFER POOL *gbpname* IS COMPLETE

Explanation: For a DISPLAY GROUPBUFFERPOOL command, this message signals the end of the display for group buffer pool *gbpname*.

DSNB791I THIS DB2 IS NOT CONNECTED TO *gbpname*

Explanation: This message is issued in response to a DISPLAY GROUPBUFFERPOOL command with TYPE(MCONN) and either one name or a list of names specified in the selection list. The DB2 on which the DISPLAY GROUPBUFFERPOOL command was issued is not currently connected to the specified group buffer pool. This message is issued once for each group buffer pool that does not match the "member connectivity" criterion.

System action: The display is terminated for this group buffer pool. Command processing continues, if there are other group buffer pool names to process.

DSNB792I NO DB2 IN THE GROUP IS CONNECTED TO *gbpname*

Explanation: This message is issued in response to a DISPLAY GROUPBUFFERPOOL command with TYPE(GCONN) and either one name or a list of names specified in the selection list. No DB2 in the group is currently connected to the specified group buffer pool. This message is issued once for each group buffer pool that does not match the "group connectivity" criterion.

System action: The display is terminated for this group buffer pool. Command processing continues, if there are other group buffer pool names to process.

DSNB793I DELETE NAME LIST = *s5* READ CASTOUT STATISTICS = *s6* DELETE NAME = *s7* OTHER ASYNCHRONOUS GBP REQUESTS = *s8*

Explanation: This message is displayed as a result of the DISPLAY GROUPBUFFERPOOL command with the MDETAIL option if the group buffer pool is duplexed. It lists the statistics on requests to the secondary group buffer pool that are related to castout processing and to garbage collection processing.

s5 The number of Delete Name List requests to delete a set of pages from the secondary group buffer pool that have just been cast out from the primary. If the group buffer pool has been duplexed for the entire reporting interval, then this number should be roughly equal to the number of Unlock Castout requests to the primary (indicated in message DSNB796I).

s6 The number of Read Castout Statistics requests to check for "orphaned data entries" in the secondary group buffer pool. The DB2 member that is the group buffer pool structure owner periodically issues these requests to determine whether garbage collection is necessary to clean up orphaned data entries from the secondary. Orphaned data entries are "changed" data entries in the secondary group buffer pool without a corresponding "changed" data entry in the primary. Orphaned data entries are to be expected from time to time and do not represent any kind of data inconsistency problem (because the page has already been cast out from the primary and such that the version of the page in the secondary is the same as the version of the page on DASD). Orphaned data entries can be introduced by certain transaction or system failures, or by new group buffer pool write

activity that encounters very unlikely timing windows when castout is concurrently processing.

s7 The number of Delete Name requests to delete "orphaned data entries" from the secondary group buffer pool. The DB2 member that is the group buffer pool structure owner issues these requests if it determines that garbage collection is necessary. See the description above under *s6* for a description of "orphaned data entries".

s8 The number of asynchronous CF requests sent to the secondary group buffer pool. This excludes ASYNCHRONOUS COMPLETION CHECKS as reported on the DSNB764I message.

For information about the Member Detail Report, see *DB2 Data Sharing: Planning and Administration*.

DSNB794I THIS DB2 IS NOT CONNECTED TO ANY GROUP BUFFER POOL

Explanation: This message is issued in response to a DISPLAY GROUPBUFFERPOOL command with the TYPE(MCONN) option and an asterisk (*) specified as the selection list. This message is issued when DB2 determines that there are no group buffer pools having "member connectivity" from the DB2 on which the DISPLAY GROUPBUFFERPOOL command was issued.

System action: Command processing is terminated.

DSNB795I NO DB2 IN THE GROUP IS CONNECTED TO ANY GROUP BUFFER POOL

Explanation: This message is issued in response to a DISPLAY GROUPBUFFERPOOL command with the TYPE(GCONN) option and an asterisk (*) specified as the selection list. When DB2 successfully determines (that is, no error is encountered during command processing) that there are no group buffer pools connected to any DB2 in the data sharing group.

System action: Command processing is terminated.

DSNB796I CASTOUTS PAGES CASTOUT = *c1* UNLOCK CASTOUT = *c2* READ CASTOUT CLASS = *c3* READ CASTOUT STATISTICS = *c4* READ DIRECTORY INFO = *c5*

Explanation: This is the eighth of ten messages containing local statistics, displayed as a result of the DISPLAY GROUPBUFFERPOOL command with the MDETAIL option. It lists local castout statistics for the group buffer pool. A preceding DSNB750I or DSNB751I message identifies the associated group buffer pool.

Counters:

c1 Number of pages cast out.

c2 Number of requests to unlock for castout.

c3 Number of read castout class requests.

c4 Number of read castout statistics requests.

c5 Number of read directory info requests. These requests are made during group buffer pool checkpoint.

DSNB797I OTHER INTERACTIONS REGISTER PAGE = *o1* UNREGISTER PAGE = *o2* DELETE NAME = *o3* READ STORAGE STATISTICS = *o4* EXPLICIT CROSS INVALIDATIONS = *o5* ASYNCHRONOUS GBP REQUESTS = *o6*

Explanation: This is the last of ten messages containing local statistics, displayed as a result of the DISPLAY GROUPBUFFERPOOL command with the MDETAIL option. It lists local castout statistics for the group buffer pool. A preceding DSNB750I or DSNB751I message identifies the associated group buffer pool.

Counters:

o1 Number of requests to the coupling facility to register interest in a single page.

o2 Number of requests to the coupling facility to remove the registered interest in a single page.

o3 Number of requests to the coupling facility to delete the directory entries associated with a set of pages.

o4 Number of requests to the coupling facility to read the statistics counters associated with the structure. DB2 generally issues this request once per group buffer pool checkpoint, and once per GDETAIL request on the DB2 DISPLAY GROUPBUFFERPOOL command.

o5 Number of explicit cross-invalidations for pages in the group buffer pool.

o6 Number of coupling facility requests to the primary group buffer pool that were converted from synchronous to asynchronous.

DSNB798I LAST GROUP BUFFER POOL CHECKPOINT*timestamp* MINIMUM CLEAN-TO-DIRTY LRSN = *lrsn1* MINIMUM MEMBER RESTART/REDO LRSN = *lrsn2*, *member-name1* STRUCTURE OWNER = *member-name2*

Explanation: This message is displayed in response to the DISPLAY GROUPBUFFERPOOL command. It is issued only if DB2 determines, by successfully querying the MVS CFRM policy, that the group buffer pool is allocated. It is immediately preceded by message DSNB759I. A preceding DSNB750I message identifies

the associated group buffer pool.

timestamp
The date and time of the last group buffer pool checkpoint.

lrsn1 The recovery log record sequence number (LRSN) that was recorded by the last group buffer pool checkpoint that reflects the oldest changed page clean-to-dirty LRSN value at the time of the checkpoint. If the group buffer pool were to fail, *lrsn1* is used by DB2 as input to determine the starting points for the merged log scan to recover the page sets and partitions that are entered into group buffer pool recovery pending (GRECP) status. You can do more detailed monitoring of the group buffer pool checkpoint processing with IFCID 0261.

lrsn2 The minimum member-level write-pending (restart/redo) LRSN that was recorded by the last group buffer pool checkpoint. *member-name1* is the DB2 member associated with the minimum restart/redo LRSN. Group buffer pool checkpoint records a restart/redo LRSN for each member - *lrsn2* reflects only the single minimum value across all members. If the group buffer pool were to fail, DB2 uses each member's restart/redo LRSN in conjunction with *lrsn1* to determine the starting points for the merged log scan to recover the page sets and partitions that are entered into group buffer pool recovery pending (GRECP) status. You can do more detailed monitoring of the group buffer pool checkpoint processing with IFCID 0261.

member-name2
The member name of the group buffer pool structure owner. The structure owner's responsibilities include the following:
- Issue the group buffer pool checkpoint.
- Detect the GBPOOLT castout threshold and initiate castout if this threshold has been reached (by notifying the pageset or partition castout owners).
- Coordinate the damage assessment process in case the group buffer pool fails.
- Coordinate the rebuild process for the group buffer pool.

'UNKNOWN' is displayed if DB2 is not able to determine which member is the structure owner for the group buffer pool. This should only happen in an unexpected error situation. See the message accompanying that error.

System action: Processing continues.

DSNB799I SECONDARY GBP ATTRIBUTES ALLOCATED SIZE = *ssize* KB VOLATILITY STATUS = *svolstat* CFNAME = *cfname* CFLEVEL - OPERATIONAL = *cflevel1* CFLEVEL - ACTUAL = *cflevel2* NUMBER OF DIRECTORY ENTRIES = *sdir#* NUMBER OF DATA PAGES = *sdata#*

Explanation: For a DISPLAY GROUPBUFFERPOOL command, this message gives the attributes of the secondary group buffer pool if the group buffer pool is duplexed. It is displayed only if the group buffer pool is duplexed and if DB2 was successful in querying the MVS CFRM administrative policy. This message is not displayed if the group buffer pool is in a transition state between simplexing and duplexing.

ssize The actual allocated size (in kilobytes) of the secondary group buffer pool coupling facility structure.

svolstat The volatility status of the secondary group buffer pool:

VOLATILE
The structure is volatile.

NON-VOLATILE
The structure is non-volatile.

cfname The name of the coupling facility in which the group buffer pool is allocated. If the group buffer pool is duplexed, this is the coupling facility name associated with the primary group buffer pool.

cflevel1 The coupling facility level from a DB2 functionality perspective for the coupling facility in which the secondary group buffer pool is allocated.

cflevel2 The coupling facility level for the coupling facility in which the secondary group buffer pool is allocated.

sdir# Actual number of directory entries allocated for the secondary group buffer pool.

sdata# Actual number of data pages allocated for the secondary group buffer pool.

DSNB800I DIRECTORY ENTRY TO DATA PAGE RATIO SET TO *value* FOR *gbpname*. IT WILL TAKE EFFECT ON THE NEXT REALLOCATION

Explanation: This message is issued in response to an ALTER GROUPBUFFERPOOL command that requests a change to the directory entry to the data page ratio for a group buffer pool. It is also issued for the first ALTER GROUPBUFFERPOOL command for a group buffer pool, even when no ratio value is specified on the command (in this case, the default ratio value is being used). The new ratio takes effect the next time

the coupling facility cache structure is allocated.

The directory entry to data page ratio determines the number of directory entries and the number of data pages that are allocated in the coupling facility cache structure for the group buffer pool. If the coupling facility structure is not allocated, the first connect from any DB2 causes allocation of the structure and the new directory entry to data page ratio to take effect.

If the structure size can be determined from the MVS CFRM administrative policy, message DSNB801I is issued to indicate an estimate of the number of directory and data entries that are allocated based on the new ratio and the current size of the coupling facility structure. Otherwise, message DSNB812I is issued.

System action: Processing continues.

System programmer response: Use the -DISPLAY GROUPBUFFERPOOL command to display the number of allocated directory and data entries.

DSNB801I MAX STRUCTURE SIZE INDICATED IN MVS CFRM POLICY = *size* KB ESTIMATED NUMBER OF DIRECTORY ENTRIES = *dir#* ESTIMATED NUMBER OF DATA PAGES = *data#*

Explanation: This message is issued in response to an ALTER GROUPBUFFERPOOL command after the directory entry to data page ratio is successfully updated. It is also issued for the first ALTER GROUPBUFFERPOOL command for a group buffer pool, even when no ratio value is specified on the command (in this case, the default ratio value is used). The group buffer pool is identified in message DSNB800I, which immediately precedes this message.

Because the actual ratio change does not take affect until the group buffer pool is reallocated, this message shows the estimated number of directory entries and data pages that are allocated based on the new ratio and the maximum size of the coupling facility structure, as indicated in the active CFRM policy definition. The actual number of entries that are allocated varies based on the amount of coupling facility storage that is available at the time the allocation is attempted, and on the amount of storage overhead necessary for the coupling facility control structures. Because DB2 cannot determine these variables, the estimated number of directory and data entries displayed in this message assumes that the entire storage allocation for the coupling facility cache structure will be used for directory and data entries, which is not the case.

To determine the actual number of directory and data entries that are allocated for this group buffer pool, you must cause the buffer pool to be reallocated or rebuilt, as described in *DB2 Data Sharing: Planning and Administration*, then issue a DISPLAY GROUPBUFFERPOOL command.

System action: Processing continues.

DSNB802I *"value"* IS NOT A VALID PARAMETER FOR 'RATIO'

Explanation: An illegal value for the RATIO parameter was specified for an ALTER GROUPBUFFERPOOL command. The RATIO value can be up to 6 characters, and must be a decimal number from 1.0 to 255 (inclusive). For values less than 25, any digits after the first decimal place are ignored. If the value for the RATIO parameter is greater than or equal to 25, any digits after the decimal point are ignored. Informational message DSNB803I is displayed if the system ignores any precision that exceeds the limits of the RATIO parameter.

System action: The request is rejected.

System programmer response: Reissue the ALTER GROUPBUFFERPOOL command, and specify an acceptable value for RATIO.

DSNB803I RATIO IS TRUNCATED TO *value*

Explanation: The precision specified on the RATIO parameter on an ALTER GROUPBUFFERPOOL command was greater than allowed. For decimal numbers less than 25, any digits after the first decimal place are ignored: for example, 5.67 is treated as 5.6. For decimal numbers greater than or equal to 25, any digits after the decimal point are ignored: for example, 25.3 is treated as 25.

System action: Processing continues.

DSNB804I CLASS CASTOUT THRESHOLD SET TO *value*% FOR *gbpname*

Explanation: This message is issued in response to an ALTER GROUPBUFFERPOOL command that requests a change to the class castout threshold value for a group buffer pool. It is also issued for the first ALTER GROUPBUFFERPOOL command for a group buffer pool, even when no threshold value is specified on the command (in this case, the default class castout threshold value is being used).

System action: Processing continues.

DSNB805I *value* IS NOT A VALID PARAMETER FOR 'CLASST'

Explanation: An illegal value was specified on the CLASST (class castout threshold) keyword on an ALTER GROUPBUFFERPOOL command. The CLASST value must be an integer between 0 and 90, inclusive.

System action: The request is rejected.

System programmer response: Reissue the -ALTER GROUPBUFFERPOOL command, specifying an

acceptable value for CLASST.

DSNB806I GROUP BUFFER POOL CASTOUT THRESHOLD SET TO *value*% FOR *gbpname*

Explanation: This message is issued in response to an ALTER GROUPBUFFERPOOL command that successfully changes the group buffer pool castout threshold value for a group buffer pool. It is also issued for the first successful ALTER GROUPBUFFERPOOL command for a group buffer pool, even when no threshold value is specified on the command (in this case, the default group buffer pool castout threshold value is being used).

System action: Processing continues.

DSNB807I *value* IS NOT A VALID PARAMETER FOR 'GBPOOLT'

Explanation: An ALTER GROUPBUFFERPOOL command was issued with an illegal value specified on the GBPOOLT (group buffer pool castout threshold) keyword. The GBPOOLT value must be an integer between 0 and 90, inclusive.

System action: The request is rejected.

System programmer response: Reissue the -ALTER GROUPBUFFERPOOL command, specifying an acceptable value for GBPOOLT.

DSNB808I GROUP BUFFER POOL CHECKPOINT INTERVAL SET TO *value* MINUTES FOR *gbpname*

Explanation: This message is issued in response to an ALTER GROUPBUFFERPOOL command that successfully changes the group buffer pool checkpoint interval value for a group buffer pool. It is also issued for the first successful ALTER GROUPBUFFERPOOL command for a group buffer pool, even when no interval value is specified on the command (in this case, the default group buffer pool checkpoint interval value is being used).

System action: Processing continues.

DSNB809I *value* IS NOT A VALID PARAMETER FOR 'GBPCHKPT'

Explanation: An ALTER GROUPBUFFERPOOL command was issued with an illegal value specified on the GBPCHKPT (group buffer pool checkpoint) keyword. The GBPCHKPT value must be an integer between 1 and 999999, inclusive.

System action: The request is rejected.

System programmer response: Reissue the -ALTER GROUPBUFFERPOOL command, specifying an acceptable value for GBPCHKPT.

DSNB812I SIZE ESTIMATES ARE NOT AVAILABLE FOR THE GROUP BUFFER POOL MVS IXCQUERY RETURN CODE = *return code* MVS IXCQUERY REASON CODE = *reason code*

Explanation: The directory entries and data pages estimates for the group buffer pool, identified on the preceding DSNB800I message, could not be computed because IXCQUERY was unable to obtain the backing structure's size information from the MVS CFRM administrative policy definition. This can happen if an IXCQUERY processing error occurred, or if the structure was not defined in the CFRM policy. The backing structure for the group buffer pool must be defined in the CFRM policy before the group buffer pool can be allocated. The exact reason can be determined from the *return code* and the *reason code*. For an explanation of the IXCQUERY codes, see *MVS/ESA Programming: Sysplex Services Reference*.

System action: The command completes successfully, but without giving the estimates.

DSNB813I *csect-name* ALTER NOTIFICATION FAILED

Explanation: An ALTER GROUPBUFFERPOOL command failed to propagate the request for castout threshold or checkpoint interval change to other DB2s in the data sharing group. This error is triggered by an error encountered by the IRLM notification process. As a result, different DB2s in the group might have different values for the castout threshold or checkpoint interval. The RATIO change does not use the notification process and is, therefore, not affected by such a failure.

System action: Processing continues.

System programmer response: Reissue the -ALTER GROUPBUFFERPOOL command with the same parameters. If the problem persists, it indicates a deeper problem, such as inadequate space for the locking structures.

DSNB814I *csect-name* NO CHANGES REQUESTED ON COMMAND

Explanation: An ALTER GROUPBUFFERPOOL command without any parameters was issued for an existing group buffer pool. If an ALTER GROUPBUFFERPOOL command without any parameters is issued for a group buffer pool that does not yet exist, a group buffer pool definition with default parameters values is created.

System action: The command is successful and processing continues.

DSNB815I **PAGE P-LOCK LOCK/UNLOCK, P-LOCK REQUESTS FOR SPACE MAP PAGES =** *nn* **P-LOCK REQUESTS FOR DATA PAGES =** *nn* **P-LOCK REQUESTS FOR INDEX LEAF PAGES =** *nn* **P-LOCK UNLOCK REQUESTS =** *nn*

Explanation: This message is issued in response to a DISPLAY GROUPBUFFERPOOL command. This message reports the physical lock (P-lock) status at the Group Buffer Pool level. The accounting of these P-locks is intended for use in monitoring and tuning database performance, especially in a data-sharing environment. For more information about the DISPLAY GROUPBUFFERPOOL command, please see the DB2 Command Reference.

System action: Processing Continues.

DSNB816I **PAGE P-LOCK SUSPENSIONS/ NEGOTIATIONS, P-LOCK SUSPENSIONS FOR SPACE MAP PAGES =** *nn* **P-LOCK SUSPENSIONS FOR DATA PAGES =** *nn* **P-LOCK SUSPENSIONS FOR INDEX LEAF PAGES =** *nn* **P-LOCK NEGOTIATIONS FOR SPACE MAP PAGES =** *nn* **P-LOCK NEGOTIATIONS FOR DATA PAGES =** *nn* **P-LOCK NEGITIATIONS FOR INDEX LEAF PAGES =** *nn*

Explanation: This message is issued in response to a DISPLAY GROUPBUFFERPOOL command. This message reports the physical lock (P-lock) status at the Group Buffer Pool level. These P-lock statistics are intended for use in monitoring and tuning database performance, especially in a data-sharing environment. For more information about the DISPLAY GROUPBUFFERPOOL command, please see the DB2 Command Reference.

System action: Processing Continues.

DSNB890I *command-verb gbp-kwd* **COMMAND IS ONLY VALID FOR DATA SHARING**

Explanation: An attempt was made to issue a GROUPBUFFERPOOL command on a DB2 that was not initialized to participate in data sharing.

command-verb
The name of the command attempted: ALTER, or DISPLAY

gbp-kwd
The keyword GROUPBUFFERPOOL or GBPOOL

System action: The command request is rejected.

DSNB891I *name* **IS NOT A VALID GROUP BUFFER POOL OR COUPLING FACILITY STRUCTURE NAME**

Explanation: An invalid name was specified on an ALTER GROUPBUFFERPOOL or DISPLAY GROUPBUFFERPOOL command.

name Either the DB2 group buffer pool name or the backing coupling structure name

You must either specify the DB2 group buffer pool name (*gpbname*) or the backing coupling facility structure name (*strname*). Acceptable values for *gbpname* are GBP0, GBP1, ... , GBP49, GBP32K, GBP32K1, ... , GBP32K9. The *strname* has the following format:

```
groupname_gbpname
```

The *groupname* is the DB2 data sharing group name. The underscore (_) is used as a delimiter between the *groupname* and the *gbpname*.

System action: The request is rejected.

System programmer response: Reissue the command specifying a valid group buffer pool name.

DSNB892I *grpname* **IS NOT A VALID GROUP NAME FOR THIS DB2**

Explanation: The group name specified as part of the full coupling facility cache structure name on an ALTER GROUPBUFFERPOOL or DISPLAY GROUPBUFFERPOOL command is incorrect. On the commands, you can specify either the correct 16-byte coupling facility cache structure name or the DB2 group buffer pool name. The coupling facility cache structure name is a 16-character name specified as follows:

```
groupname_gbpname
```

The *gbpname* can be GBP0, GBP1, ..., GBP49 GBP32K, GBP32K1, ..., GBP32K9. The *groupname* is the DB2 data sharing group name. The underscore (_) is used as a delimiter between the *groupname* and the *gbpname*. The coupling facility cache structure names are left-justified with a blank as the rightmost padding character. DB2 adds the padding blanks if they are not specified on the command.

System action: The request is rejected.

System programmer response: Reissue the command, specifying a valid group buffer pool name.

Chapter 4. DSNE... messages

In the TSO attachment facility, the user can control whether message IDs are displayed. To see them, type PROFILE MSGID. To suppress them, type PROFILE NOMSGID.

Trace messages associated with the TSO attachment facility (including DSN and DB2I) are not listed in this documentation. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

All DSN command processor trace messages have the same message number: DSNET20I.

DSNET20I

Explanation: This is one of many possible trace messages bearing this message number. Complete documentation of all of the TSO attachment facility trace messages bearing this message number is found in Part 8 of *DB2 Diagnosis Guide and Reference*.

DSNE004E PROBABLE SUBSYSTEM ERROR DETECTED IN CSECT csect-name, SUBSYSTEM FUNCTION service, RETCODE retcode, REASON CODE reason-code

Explanation: A subsystem error has occurred. The function identified in the message as 'service' is either an MVS or a DB2 service routine. The error was discovered in the 'csect-name' indicated in the message. Refer to the appropriate MVS publications for a description of the return and reason codes.

System action: This message is sometimes followed by an intentional system abend. In other instances, the DSN command processor will terminate the current subcommand and attempt to continue by issuing another DSN prompt.

User response: Run the job again with a SYSUDUMP dump data set allocated and DSN tracing set to 123. Collect the DSN trace stream. If an abend occurs, collect the dump.

DSNE005A EXECUTION IS INTERRUPTED. ENTER C TO CANCEL, OR ANY OTHER REPLY TO RESUME THE subcmd SUBCOMMAND

Explanation: The Attention key was pressed. Entering a 'C' will cancel the DSN subcommand that is currently executing. If the user enters a question mark, message DSNE006A will appear. Any other reply will cause execution of the subcommand that was in progress to continue.

If the DSN prompt was issued just before the Attention key was pressed, 'subcmd' may be either 'DSN' or the previously invoked subcommand.

User response: Enter a 'C' to cancel the DB2 subcommand currently executing. The DSN prompt will appear on the terminal. Enter anything else (except another attention) to resume execution.

DSNE006A C WILL TERMINATE THE SUBCOMMAND WITH UNCOMMITTED CHANGES BACKED OUT

Explanation: Enter a 'C' to terminate the subcommand. No uncommitted changes are made to the database. Any other reply causes subcommand execution to continue.

User response: Enter a 'C' to terminate the subcommand. Control returns to DSN, and the DSN prompt appears on the terminal. This is a second-level help message associated with message DSNE005A.

DSNE008I REPLY WAS xxxxx, EXECUTION RESUMED

Explanation: Execution was suspended by a user ATTENTION and then was resumed, because the string specified in the message as 'xxxxx' was entered.

System action: Control returns to the interrupted session.

DSNE009I REPLY WAS C, EXECUTION TERMINATED

Explanation: The response to message DSNE005A was a C. This caused the current subcommand to terminate.

System action: The current subcommand is terminated. The DSN prompt appears on the terminal.

© Copyright IBM Corp. 1983, 2008

User response: You can now enter another DSN subcommand.

DSNE010I SUBCOMMAND subcmd CANCELED BY ATTENTION

Explanation: The subcommand specified in the messages was canceled because the user pressed the attention key and then entered a 'C'.

System action: The current subcommand is terminated. The DSN prompt appears on the terminal.

User response: A new subcommand can be entered.

DSNE011I DSN COMMAND CANCELED BY ATTENTION

Explanation: The user of the DSN command processor has pressed attention. Since this occurred before DSN was fully initialized, DSN terminates. If DSN had been fully initialized, it would have prompted the user to continue or cancel.

System action: The DSN command processor terminates. Control returns to TSO or the calling CLIST.

User response: Restart DSN.

DSNE100I ssid NOT OPERATIONAL, RETRY COUNT IS ZERO

Explanation: The subsystem (specified in the message as 'ssid') was not active, and no retries remain to be executed. No more connection attempts are made. The operator should be notified to start this specific subsystem, and the DSN command should be resubmitted. Issue the DSN command with the RETRY parameter set greater than zero.

System action: Control returns to TSO.

User response: Notify the operator to start the subsystem.

Operator response: Start the subsystem with the 'ssid' specified in the message.

DSNE101I ssid NOT OPERATIONAL, WILL RETRY nnn MORE TIMES

Explanation: The requested DB2 subsystem 'ssid' was not started or is not available. Connection to it will be attempted 'nnn' more times. This message is issued every 30 seconds (and each time 'nnn' is decreased by 1) until the connection is made or 'nnn' is zero.

System action: DSN repeats the connection attempt until the connection is made or until the retry count is decreased to zero.

User response: Notify the operator start DB2.

DSNE102I YOU ARE NOT AUTHORIZED TO USE DB2 SUBSYSTEM ssid

Explanation: The authorization ID is not authorized to use subsystem 'ssid'.

System action: Control returns to TSO.

User response: Contact the system programmer, or system administrator to arrange for the required RACF or equivalent security system authorization.

DSNE103E MAIN STORAGE REQUEST FROM csect-id FAILED

Explanation: A request for main storage failed.

System action: The TSO attachment facility is ready for more input.

User response: Increase the address space size. This can be done as part of the logging on process.

DSNE104E RETRY COUNT OF number NOT VALID, SET TO 120

Explanation: An invalid 'RETRY' parameter was entered on the DSN command. It has been set to 120. Valid retry counts are integer numbers from 0 to 120.

System action: Retry count is modified, and DSN execution continues.

User response: The next time you invoke DSN, use a RETRY count of 0 to 120.

DSNE105E SUBSYSTEM ssid NO LONGER OPERATIONAL

Explanation: The DB2 subsystem is down. It must be restarted before the user can continue.

System action: Control returns to TSO attachment facility.

User response: Before you can continue, DB2 must be restarted. You must also reissue the DSN command. ISPF users should restart their jobs.

DSNE106E PLAN plan-id NOT AUTHORIZED FOR SUBSYSTEM ssid AND AUTH-ID auth-id

Explanation: The requested combination of plan-id, subsystem-id, and user-id is not currently authorized.

System action: The TSO attachment facility is ready for more input.

User response: Notify the system programmer or system administrator to obtain proper authorization.

System programmer response: The system administrator for the DB2 subsystem can arrange the necessary authorization.

DSNE107E SUBSYSTEM RESOURCE NEEDED FOR PLAN plan-name, AUTH ID auth-id, AND SUBSYSTEM subsys IS NOT AVAILABLE

Explanation: The DB2 subsystem was unable to allocate a necessary resource for the requested combination of plan-id, user-id, and subsystem-id. This message is accompanied by DSNE108E. An attempt to establish connection with the DB2 subsystem has failed.

System action: The TSO attachment facility is ready for more input.

User response: Arrange to obtain the needed resource, and resubmit the subcommand. See the explanation for message DSNE108E, which accompanies this message.

DSNE108E FEEDBACK - REASON CODE reason-code TYPE= type RESOURCE NAME - 44 characters

Explanation: This message accompanies message DSNE107E or DSNE114E to explain the problem more clearly. RESOURCE NAME is the name of the unavailable resource. The reason code is in hexadecimal format.

System action: The TSO attachment facility is ready for more input.

User response: Arrange for the missing resource to be available, and resubmit your job.

Problem determination: The reason the resource is unavailable is identified by a reason code. The reason codes that appear in this message are described in *DB2 Codes*.

TYPE and RESOURCE NAME identify the resource that is currently unavailable. The TYPE code identifies the type, as well as the format and content, of the RESOURCE NAME. Refer to Table 2 in Appendix A, "Problem determination," on page 659 for an explanation of resource type codes.

DSNE109E NO INPUT TO BIND, SUBCOMMAND TERMINATED

Explanation: No LIBRARY parameter was specified on the BIND subcommand. No DBRMLIB was allocated.

System action: The TSO attachment facility is ready for more input.

User response: Specify the LIBRARY parameter, and resubmit the BIND subcommand. Alternatively, you could issue an ALLOCATE command for the DDNAME DBRMLIB and then resubmit the BIND subcommand as originally written.

DSNE110E ssid NOT VALID SUBSYSTEM ID, COMMAND TERMINATED

Explanation: There is no subsystem id with the name identified in the message as 'ssid'. The subsystem identifier you specify (or default to) on the DSN command must exist. This message is the result of specifying (or defaulting to) a nonexistent subsystem identifier. For information on the SYSTEM parameter of the DSN command, refer to *DB2 Command Reference*.

System action: The DSN processor ends, and control returns to TSO.

User response: Reissue the command with a valid SYSTEM parameter.

DSNE111E INSTALLATION EXIT REJECTED DSNAME data-set-name

Explanation: A local installation-supplied exit has rejected the data set name identified in the message as 'data-set-name'.

System action: The TSO attachment facility is ready for more input.

User response: Either correct the data set name specified, or arrange to authorize the original data set name. Resubmit the request.

DSNE112E userid STILL SIGNED ON TO SUBSYSTEM ssid

Explanation: An attempt was made to sign on to DB2 when the user was already signed on.

System action: Control returns to TSO.

User response: Log off and log on again.

DSNE113E NOT ABLE TO LINK TO MODULE modname IN DSNAME data-set-name

Explanation: A DB2 subsystem error occurred during an attempt to link to module 'modname' in data set 'data-set-name'.

System action: The TSO attachment facility is ready for more input.

User response: Check that the module exists and that its name is spelled correctly. Enter the correct module name.

DSNE114E THE COMBINATION OF PLAN planid, AUTH ID auth-id, AND SUBSYSTEM ssid IS NOT VALID

Explanation: The resource in message DSNE108E, which will follow this message, is not a valid resource name.

System action: The TSO attachment facility is ready for more input.

User response: Correct the invalid combination, and resubmit the subcommand. The problem is probably caused by a user spelling error. If it is not, check with the system administrator to determine the availability of the needed resource. Your program may need to be bound. See the explanation for message DSNE108 (which accompanies this one) for more information.

DSNE117E NOT ABLE TO LOCATE membername IN DSNAME data-set-name

Explanation: The 'data-set-name' specified does not contain the member the user wanted to run.

System action: Control remains with the DSN command processor, and the DSN prompt appears on the terminal. The DSN command processor is ready for another command.

User response: Correct the PROGRAM or LIBRARY parameter, and resubmit the subcommand.

DSNE118E invalid-command NOT VALID COMMAND

Explanation: The user entered an invalid command. A typing error may be responsible. You may be trying to run a command processor that is either not available on your system or that is not part of your current LNKLST or STEPLIB concatenation.

System action: The TSO attachment facility is ready for more input.

User response: Either correct the spelling error, or rearrange your library concatenation so that you can access the required command processor. Resubmit the command.

DSNE119E data-set-name NOT PARTITIONED DATA SET

Explanation: The user tried to allocate a sequential data set when a partitioned data set was required.

System action: The TSO attachment facility is ready for more input.

User response: Check the data-set-name. The data set you specified is sequential, but you specified a member name. Remove the member name, and resubmit the job.

DSNE120E data-set-name NOT SEQUENTIAL DATA SET

Explanation: The user tried to allocate a partitioned data set when a sequential data set was required.

This message is issued by the following CSECT: DSNECP69

System action: The TSO attachment facility is ready for more input.

User response: Check the data-set-name. The data set you specified is partitioned, but you did not specify a member name. Add a member name, and resubmit the job.

DSNE121E *csect-id* **NOT ABLE TO OPEN DATA SET**

Explanation: The DB2 subsystem was unable to open the requested data set.

System action: The TSO attachment facility is ready for more input.

User response: For information about why the data set could not be opened, read the explanations of the messages that accompany this message. Correct the condition, and resubmit the job.

DSNE122E DATA SET ORGANIZATION MUST BE EITHER SEQUENTIAL OR PARTITIONED. data-set-name IS NOT

Explanation: The DB2 subsystem was unable to open data set 'data-set-name', because it is not a sequential or partitioned data set.

This message is issued by the following CSECT: DSNECP69

System action: The TSO attachment facility is ready for more input.

User response: Correct the data-set-name, or change the data set organization. Resubmit the job.

DSNE123I *data-set-name* **NOT PARTITIONED DATA SET, IT IS IGNORED**

Explanation: The data-set-named in the message is required to be partitioned. It is not and, consequently, was ignored.

System action: DSN will attempt to continue BIND processing.

User response: Correct the subcommand, and resubmit it.

DSNE124E DATA SET *data-set-name*, **MEMBER** *member-name* **IN USE, TRY LATER**

Explanation: The member contained in 'data-set-name' is currently in use. This message is normally issued when another user is updating the specified member.

This message is issued by the following CSECT: DSNECP69

System action: The TSO attachment facility is ready for more input.

User response: Wait until the member is available and reissue your DCLGEN request.

DSNE125E DATA SET *data-set-name* IN USE, TRY LATER

Explanation: The sequential data set 'data-set-name' is currently in use. This message is normally issued when another user is updating the specified data set.

This message is issued by the following CSECT: DSNECP69

System action: The TSO attachment facility is ready for more input.

User response: Wait until the data set is available and reissue your DCLGEN request.

DSNE126E THE DB2 OPERATOR IS STOPPING THE SUBSYSTEM. PLEASE END YOUR DSN SESSION

Explanation: The operator has requested that DB2 quiesce. The user should promptly finish the current task and terminate the DSN session so that this can take place. This message may be followed by message DSNE127I.

This message is issued by the following CSECT: DSNECP00

System action: No new users will be permitted to initiate DB2 sessions. Current users will be permitted to work until they terminate their connections to DB2.

User response: End your DSN session as soon as possible.

DSNE127I THE DB2 OPERATOR IS STOPPING THE SUBSYSTEM. DSN IS TERMINATING

Explanation: The operator has requested that DB2 terminate.

System action: DB2 will shut down, interrupting user activities as necessary. DSN will terminate immediately and return the user to TSO.

User response: Reinitiate your session when DB2 returns.

DSNE128I DB2 IS ABNORMALLY ENDING. DSN IS TERMINATING

Explanation: DB2 has encountered a problem from which it cannot recover.

System action: DB2 is abending. DSN terminates immediately and returns the user to TSO.

User response: Reinitiate your session when DB2 returns.

DSNE130I MAXIMUM NUMBER OF DB2 USERS REACHED. YOUR RETRY COUNT IS ZERO

Explanation: The number of concurrent connections to DB2 is restricted. The number is specified in the installation parameters in member DSN6SYSP. For TSO foreground, see parameter IDFORE. For background jobs, see parameter IDBACK. The DSN request has been denied, because the current number of DB2 connections in the system has reached the installation-specified limit.

System action: Control returns to TSO.

User response: Resubmit the DSN command when activity using DB2 has diminished. If still unsuccessful, see the system programmer about the installation parameters.

DSNE131I MAXIMUM NUMBER OF DB2 USERS REACHED. WILL RETRY nnn MORE TIMES

Explanation: The number of concurrent connections to DB2 is restricted. The number is specified in the installation parameters in member DSN6SYSP. For TSO foreground, see parameter IDFORE. For background jobs, see parameter IDBACK. The DSN request was denied because the current number of DB2 connections in the system reached the installation specified limit.

The DSN connection to DB2 is attempted 'nnn' more times. The number of retries is determined by the RETRY parameter on the DSN command. For more information, refer to *DB2 Command Reference*.

This message is issued every 30 seconds (and each time 'nnn' is decreased by 1) until the connection is made or until 'nnn' is decreased to zero.

System action: DSN repeats the connection attempt until the connection is made or the retry count is decreased to 0.

User response: If all retry attempts fail, resubmit the DSN command when DB2 activity diminishes. If still unsuccessful, see the system programmer about the installation parameters.

DSNE132I DB2 IS OPERATING IN RESTRICTED ACCESS MODE. MAINTENANCE PERSONNEL ONLY

Explanation: DB2 is not available for normal operation. It is only available to service personnel.

System action: Control returns to the caller. Your request was not processed. The DSN command processor was unable to connect to DB2.

User response: Wait until DB2 is available for general use, then resubmit your request.

DSNE133E *csect-name* **RELEASE LEVELS NOT COMPATIBLE. DSN=***release_level***, DB2=***release_level*

Explanation: The DSN command processor, perhaps running under the control of DB2I, has detected a release level incompatibility between itself and DB2. The current *release_level* of the DSN command processor code and the DB2 subsystem load modules is indicated as a string of three numeric characters as follows:
- Version
- Release
- Modification level

System action: The DSN command processor terminates.

User response: Check that your method of invoking DSN or DB2I allocates the correct libraries for DSN. If you used JCL or a TSO logon procedure to execute DSN code, check your JOBLIB and STEPLIB allocations. If you used a CLIST to execute DSN, insure that you obtained the correct version of the DSN code.See your system programmer if you invoke DSN using procedures supplied by your system programmer.

Operator response: Notify the system programmer.

System programmer response: If you are unable to determine the problem, you can trace DSN execution through DB2I or the DSN command processor.

Problem determination: Insure that coexistence is supported on both release levels (DSN and DB2). If coexistence is supported, insure that the lower level release has the correct SPE level code support that is required to coexist with the current higher level release.

For more information about coexistence and SPEs, refer to the program directory or preventative service planning (PSP) updates.

DSNE136I DB2 IS OPERATING IN RESTART LIGHT MODE. INDOUBT RESOLUTION ONLY

Explanation: The DB2 member is not available for normal operation. It is only available to commit coordinators that have indoubt units of recovery that need to be resolved with this member.

System action: Control returns to the caller. Your request was not processed. The DSN command processor was unable to connect to DB2.

User response: Wait until this DB2 is available for general use, then resubmit your request. Or, request connection to a different DB2 member in the data sharing group (for example, by using the DB2 group attach name).

DSNE231A ONE NAME IS REQUIRED TO FREE A PLAN OR A PACKAGE

Explanation: The user has attempted to use the FREE PLAN or PACKAGE panel without specifying a plan or package name.

System action: The TSO attachment facility is ready for more input.

User response: Enter the name of a plan or package, or (*) to indicate 'all'.

DSNE232A CONNECTION NAME REQUIRES ENABLE/DISABLE OF CONNECTION TYPE

Explanation: The user indicated that specific connection names were to be specified for ENABLE or DISABLE, but the associated connection type field was not updated. The ENABLE or DISABLE of connection names requires the corresponding connection type to also be specified for ENABLE or DISABLE.

System action: The TSO attachment facility is ready for more input.

User response: Enter Y (Yes) on the connection type for the associated connection names.

DSNE233A MEMBER NAME MUST NOT BE SPECIFIED MORE THAN ONCE

Explanation: The LIBRARY field containing the data set name, also contains a member name.

System action: The TSO attachment facility is ready for more input.

User response: The member name must be specified only once, and must be specified on the MEMBER field.

DSNE234A EITHER DBRM MEMBER OR INCLUDE PACKAGE LIST MUST BE SPECIFIED

Explanation: The BIND PLAN subcommand requires either a DBRM member name or a package list as input, neither was specified.

System action: The TSO attachment facility is ready for more input.

User response: Enter a DBRM member name(s), or specify that a package list is to be included. Both options are also allowed.

DSNE235A *plan/package* **NAME IS A DUPLICATE OF PRIMARY** *plan/package* **NAME**

Explanation: In addition to the primary 'plan' or 'package' name specified on the REBIND or FREE subcommand panel, the user also specified that

additional plans or packages were to be included in the list. The primary 'plan' or 'package' name was again specified to be included in the list.

System action: The duplicate of the primary 'plan' or 'package' name is automatically deleted from the list to avoid errors during subcommand processing. The TSO attachment facility is ready for more input.

User response: Complete the list of additional plans or packages.

DSNE236A SPECIFY A CONNECTION TYPE TO ENABLE OR DISABLE

Explanation: The user indicated that individual connection types were to be ENABLED or DISABLED, but no specific connection types were specified.

System action: The TSO attachment facility is ready for more input.

User response: Enter Y (YES) for each individual connection type to ENABLE, or DISABLE.

DSNE237A SPECIFY ENABLE ALL OR SPECIFIC CONNECTION TYPES

Explanation: In addition to requesting that all connection types be enabled, the user also requested that specific connection types be enabled. The enable of all connection types includes the enable of specific connection types.

System action: The TSO attachment facility is ready for more input.

User response: Specify specific connection types to be enabled, or specify that all connection types are to be enabled.

DSNE240A ISPF RETURNED AN UNEXPECTED CODE WHILE CLOSING A TABLE

Explanation: An ISPF error return code was encountered when attempting to perform a TBCLOSE for an ISPF table while processing the current subcommand panel.

System action: TSO processing for the current subcommand is terminated.

User response: Notify the system programmer.

System programmer response: Verify that the ISPF table library allocated to the DSNETBLS DD statement has sufficient space or directory entries available. Also refer to the ISPF error log for additional diagnostic information.

DSNE241A INTERNAL PROCEDURE WAS PASSED AN INVALID NUMBER OF ARGUMENTS

Explanation: A TSO Interpreter error was encountered while executing a subroutine during the processing of the current subcommand.

System action: TSO processing for the current subcommand is terminated.

User response: Notify the system programmer.

System programmer response: Notify the IBM support center that an error was encountered in the TSO SYSCALL instruction.

Problem determination: The TSO SYSCALL instruction is not passing all the parameters to a subroutine defined within the CLIST.

DSNE242A ISPF RETURNED AN UNEXPECTED CODE WHILE SETTING SEARCH ARGUMENTS

Explanation: An ISPF table for the current function does not contain the expected column or table format.

System action: TSO processing for the current subcommand is terminated.

User response: Notify the system programmer.

System programmer response: Delete the ISPF table in error. The table is recreated automatically during the next invocation of the DB2I function.

Problem determination: The format of the ISPF table was changed since the last successful use of the table. Refer to the DB2I ISPF table library allocated to the DSNETBLS DD statement, and examine the ISPF error log for additional diagnostic information.

DSNE243A ISPF RETURNED AN UNEXPECTED CODE DURING A TABLE DISPLAY

Explanation: An ISPF error was encountered when attempting to display a DB2I table panel.

System action: TSO processing for the current subcommand is terminated.

User response: Notify the system programmer.

System programmer response: Insure that the DB2I panel and panel library are established correctly. If the format of both is correct, then use ISPF option 7.2 to test the panel for additional errors.

Examine the ISPF error log for additional error information and determine the name of the panel to be displayed.

DSNE244A CANNOT SAVE CREATED TABLE, VERIFY SPACE IN TABLE LIBRARY

Explanation: An ISPF TBSAVE function returned with a condition code greater than zero.

System action: TSO processing for the current subcommand is terminated.

User response: Notify the system programmer.

System programmer response: Verify that the ISPF table library allocated to the DSNETBLS DD statement has sufficient space and directory entries available. Also refer to the ISPF error log for additional diagnostic information.

DSNE245A ISPF RETURNED AN UNEXPECTED CODE WHILE ADDING A ROW TO A TABLE

Explanation: An ISPF TBADD function returned with a condition code greater than four.

System action: The TSO attachment facility is ready for more input.

User response: Enter the DB2I CANCEL command, and notify the system programmer.

System programmer response: Verify that the user has enough memory to perform ISPF functions. Also refer to the ISPF error log for additional diagnostic information.

DSNE246A ISPF RETURNED AN UNEXPECTED CODE WHILE DELETING A ROW

Explanation: An ISPF TBDELETE function returned with a condition code greater than zero.

System action: TSO processing for the current subcommand is terminated.

User response: Enter the DB2I CANCEL command, and notify the system programmer.

System programmer response: Verify that the user has enough memory to perform ISPF functions. Also refer to the ISPF error log for additional diagnostic information.

DSNE247A ISPF RETURNED AN UNEXPECTED CODE WHILE UPDATING A ROW

Explanation: An ISPF TBPUT function returned with a condition code greater than four.

System action: TSO processing for the current subcommand is terminated.

User response: Enter the DB2I CANCEL command, and notify the system programmer.

System programmer response: Verify that the user has enough memory to perform ISPF functions. Recreate the user actions that lead to the failure. Also refer to the ISPF error log for additional diagnostic information.

DSNE248A COMMAND ENTERED CONTAINS INVALID CHARACTERS, RESPECIFY

Explanation: The command entered at the command line contains an invalid character, the command must be reentered.

System action: The TSO attachment facility is ready for more input.

User response: Correct the invalid command entry.

DSNE250A PLEASE ENTER THE COMMAND AT THE LINE WITH THE PLAN/PACKAGE NAME

Explanation: SELECT or RESET commands must be entered on the line where the PLAN/PACKAGE name is shown. This line contains the name of the PLAN/PACKAGE, while the secondary line(s) do not.

System action: The line command is rejected. The TSO attachment facility is ready for more input.

User response: Reenter the line command on the primary PLAN/PACKAGE line(s).

DSNE252A JCL WAS NOT CREATED, ERROR *errcode* **WHILE DOING** *function_name* **FUNCTION**

Explanation: Error code 'errcode' was received while doing a function 'function_name', the JCL was not created.

Severity: 20

System action: Refer to *z/OS ISPF Dialog Developer's Guide* for the cause of this error.

User response: Contact your systems programmer.

Problem determination: Use the return code/function combination to find the causes for the problem in *z/OS ISPF Dialog Developer's Guide* and correct the problem.

Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 21.

DSNE283A THE , (COMMA) OPTION IS INCOMPATIBLE WITH THE *hostlang* **LANGUAGE**

Explanation: The user tried to select the 'hostlang' APPLICATION LANGUAGE and the comma option of the DECIMAL POINT field on the DB2I defaults panel. This combination is not supported.

System action: The defaults panel is displayed again.

User response: Either change the APPLICATION LANGUAGE field or the DECIMAL POINT field. If you need the C (C/370 Compiler), then the DECIMAL

POINT field must contain a decimal point (.).

DSNE288A YOU CANNOT SPECIFY (RUN)=YES WITH CICS OR IMS

Explanation: Application programs that make CICS or IMS calls cannot be run from the TSO environment. They can, however, be precompiled, bound, compiled, and linked under TSO. The user should specify N on the RUN panel when preparing a CICS or IMS application.

System action: The system is waiting for new input to the panel.

User response: Change the RUN option to N.

DSNE291A ALL DB2 COMMANDS BEGIN WITH A DASH

Explanation: The user omitted the DB2 command recognition character that must precede all DB2 commands.

System action: The DB2 commands panel is displayed again so the user can resubmit the command.

User response: Add the DB2 command recognition character, and reenter the command.

DSNE292I DATA SET *data-set-name* NOW CONTAINS YOUR JCL

Explanation: DB2I has just created some JCL for you. The JCL now resides in data set 'data-set-name'. You can now edit this data, if desired. The TSO SUBMIT command will start the job running in background.

System action: The system is waiting for your next request.

User response: You can start another DB2I job or leave the panel.

DSNE293A MEMBER NAME IS REQUIRED. YOUR PROGRAM MUST BE IN A PDS

Explanation: The user either has attempted to run a program that is not in a partitioned data set or has forgotten to specify a member name.

System action: The RUN panel will be displayed so that the user can specify a member name and try the run again.

User response: Specify a member name, and rerun the program.

DSNE294I SYSTEM RETCODE=*hex-code* USER OR DSN RETCODE=*decimal-code*

Explanation: A DB2I function has just completed. The return codes for that function are returned in this message. The variable 'hex-code' is a 3-digit hexadecimal number; the variable 'decimal-code' is a 3-digit decimal number. Refer to the appropriate MVS publication for explanations of the return codes.

System action: DB2I is ready for another execution cycle unless other messages appeared indicating that DB2 is unavailable.

User response: You may have to change input parameters to correct unsatisfactory completion codes.

DSNE295A YOU CANNOT DISPLAY AND RESTART A UTILITY AT THE SAME TIME

Explanation: You have set options on the DB2I Utilities panel that call for conflicting actions. You can either DISPLAY the status of a utility, or you can RESTART a utility, but you cannot do both.

System action: The system is waiting for new input to the panel.

User response: Either change the RESTART option to NO, or select a different FUNCTION.

DSNE296A YOU CANNOT TERMINATE AND RESTART A UTILITY AT THE SAME TIME

Explanation: You have set conflicting options on the DB2I Utilities panel. You can either TERMINATE a utility, or you can RESTART a utility, but you cannot do both.

System action: The system is waiting for new input to the panel.

User response: Either change the RESTART option to NO, or select a different FUNCTION.

DSNE297A ENTER DATA SET NAME: COPYDSN OR RCPYDSN1

Explanation: You have not specified a data set name for the COPY utility. A data set name for the local/current site (COPYDSN), and/or the recovery site (RCPYDSN1) is required.

System action: The system is waiting for new input to the panel.

User response: Enter a data set name for COPYDSN, and/or RCPYDSN1.

DSNE298A ERROR DURING EDIT: CLIST *data set name*

Explanation: Errors were encountered while attempting to change the online book data set names in the DB2 CLIST member DSNEMC01. No data set names were changed.

System action: The system is waiting for new input to the panel.

User response: Ensure that the CLIST data set name is correct and contains the DSNEMC01 CLIST member to be changed. This CLIST data set must also be available to the TSO user ID in WRITE mode. This error will continue to be displayed on the panel until the error is resolved or the user exits the panel (END).

System programmer response: If you are unable to determine the problem, you can trace the DB2 CLIST execution while displaying this panel.

Problem determination: Examine the CLIST trace output for any TSO/ISPF nonzero return codes encountered during CLIST execution.

If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNE340I INTERNAL ERROR – ILLEGAL TYPE IN MODULE *module-name*

Explanation: An internal DB2 problem has occurred. The named module has detected a nonexistent SQLTYPE.

User response: Collect the SPUFI trace and ISPF log, and notify either the system programmer or the database administrator.

You may have to rerun the failing job after you have turned on trace. This will enable you to send trace messages to the ISPF log.

System programmer response: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNE345I WARNING: DB2 DATA CORRUPTION CAN RESULT FROM THIS SPUFI SESSION BECAUSE THE CCSID USED BY THE TERMINAL IS NOT THE SAME AS THE CCSID USED BY SPUFI. TERMINAL CCSID *terminal-ccsid***. SPUFI CCSID:** *spufi-ccsid***. NOTIFY THE DB2 SYSTEM ADMINISTRATOR.**

Explanation: The current SPUFI session was initiated on a TSO terminal that uses a coded character set identifier (CCSID) of *terminal-ccsid*. However, SPUFI was bound on the local DB2 server using a CCSID of *spufi-ccsid*. This discrepancy can lead to unexpected data conversion, affecting any characters that do not map to the same code point in the two CCSIDs.

If *terminal-ccsid* is blank, then the terminal code page and character set cannot be queried or they are not supported by ISPF. Possible reasons why the terminal code page and character set cannot be queried include:

• Hardware limitation such as use of older terminals

• Improper configuration of the terminal emulator software

• Improper BTAM or control unit setup

A blank *spufi-ccsid* indicates that SPUFI might have received an unexpected SQLCODE when it attempted to query the current application encoding scheme. Diagnostics for that SQLCODE appear directly after the DSNE345I message in the message warning panel.

System action: SPUFI waits for the user to indicate whether to continue or quit.

User response: Notify the system programmer. Depress the END key to quit or the ENTER key to continue. **Important:** Continuing can result in corruption of data stored by DB2.

System programmer response: On the local DB2, SPUFI expects to process only EBCDIC character data encoded in the CCSID of *spufi-ccsid*. Ensure that the user's TSO terminal configuration is updated to use a CCSID of *spufi-ccsid*. Verify the data integrity of all tables modified during the SPUFI session and follow standard recovery procedures. Contact IBM for guidance if widespread or long term data corruption is evident (for example, if many users report this message when using SPUFI).

Special considerations:

Blank CCSIDs

As noted previously, a blank *terminal-ccsid* setting occurs when the terminal code page and character set cannot be queried or they are not supported by ISPF. Check for the following situations:

• If the message occurs when using a terminal emulator, verify that the emulator software is configured with a CCSID setting.

• Check the VTAM setup: The VTAM logmode entry used to define the session characteristics for the terminal must have the EDS (extended data stream) bit turned 'on' in the PSERVIC (that is, pservic=xx80xx..). For more information, consult *OS/390 ISPF Planning and Customization* and the *z/OS SNA Network Implementation Guide*.

• Older terminal hardware such as the IBM 327x series do not support the EDS, so a CCSID cannot be determined. You can instruct end users of such devices to specify in the SPUFI defaults panel that the DSNE345I message should be suppressed. Note that suppression applies only when the terminal CCSID is blank. You cannot use this field to suppress DSNE345I when *terminal-ccsid* is non-blank.

For more information about *terminal-ccsid* and *spufi-ccsid*, consult the *DB2 Installation Guide*.

A blank *spufi-ccsid* indicates that SPUFI probably
received an unexpected SQLCODE when it attempted
to query the current application encoding scheme.
Check the diagnostics for the SQLCODE that is
displayed directly after the DSNE345I message, and
then respond to the diagnostics accordingly.

**# DSNE346I WARNING: ALL SHIFT-IN AND
SHIFT-OUT CHARACTERS IN THE
'FOR BIT DATA' COLUMNS IN THE
OUTPUT ABOVE HAVE BEEN
REPLACED BY THE SUBSTITUTION
CHARACTER '.'**

Explanation: The current SQL query returned one or
more columns that have a data type of CHAR FOR BIT
DATA, VARCHAR FOR BIT DATA, or both. Also, the
current SPUFI session indicates that a graphic character
set is being used. To prevent the possible corruption of
outputted data elsewhere in the report, all shift-in (code
point x'0E') and shift-out (code point x'0F') characters
will be replaced with a substitution character of '.'
(code point x'4B').

System action: SPUFI processing continues.

User response: To avoid the substitution character for
shift-in and shift-out data, and in order to see all FOR
BIT DATA in hexadecimal format, go to the SPUFI
defaults panel and change the setting of the FOR BIT
DATA field from ASIS to HEX.

DSNE351A EDIT PROCESSING OPTION MUST BE Y (YES) OR N (NO)

Explanation: The user specified an edit option other than 'YES', 'NO', or the abbreviation 'Y' or 'N'. A valid user response determines whether SPUFI invokes the ISPF editor to allow the user to edit the input data set.

System action: SPUFI will wait until the user enters a valid response.

User response: Type in a 'Y' or an 'N', and press 'ENTER'.

DSNE352A EXECUTE PROCESSING OPTION MUST BE Y (YES) OR N (NO)

Explanation: The user specified an execute option other than 'YES', 'NO', or the abbreviation 'Y' or 'N'. A valid user response determines whether SPUFI executes the SQL statements in the user input data set.

System action: SPUFI waits until the user enters a valid response.

User response: Type in a 'Y' or an 'N', and press 'ENTER'.

DSNE353A BROWSE PROCESSING OPTION MUST BE Y (YES) OR N (NO)

Explanation: The user specified a browse option other than 'YES', 'NO', or the abbreviation 'Y' or 'N'. A valid user response determines whether SPUFI invokes ISPF BROWSE to allow the user to browse the output data set.

System action: SPUFI will wait until the user enters a valid response.

User response: Type in a 'Y' or an 'N', and press 'ENTER'.

DSNE354A AUTOCOMMIT PROCESSING OPTION MUST BE Y (YES) OR N (NO)

Explanation: The user specified an AUTOCOMMIT option other than 'YES', 'NO', or the abbreviation 'Y' or 'N'. A valid user response determines whether SPUFI automatically commits all the changes to the database subsystem. This was encountered as a result of executing the user's SQL statement(s). The commit is done at the end of the processing of an input data set, if it is requested and if there has been no SQL execution error.

System action: SPUFI waits until the user enters a valid response.

User response: Type in a 'Y' or an 'N', and press 'ENTER'.

DSNE355A CHANGE DEFAULTS PROCESSING OPTION MUST BE Y (YES) OR N (NO)

Explanation: The user specified a change-default option other than 'YES', 'NO', or the abbreviation 'Y' or 'N'. A valid user response determines whether SPUFI displays the current-session SPUFI defaults and allows the user to change the defaults.

System action: SPUFI waits until the user enters a valid response.

User response: Type in a 'Y' or an 'N', and press 'ENTER'.

DSNE359A OUTPUT DATA SET NAME REQUIRED FOR EXECUTE OR BROWSE

Explanation: The user did not specify the name of the output data set.

System action: SPUFI waits until the user specifies an output data set.

User response: Type in the correct entry, and press 'ENTER'.

DSNE360A THE OUTPUT DATA SET MUST RESIDE ON DASD

Explanation: The user has attempted to send the output of SPUFI to a SYSOUT data set. He must enter a generic device class that contains only DASD devices.

System action: SPUFI waits until the user specifies a valid output device class.

User response: Type in the correct entry, and press 'ENTER'.

DSNE361I SPUFI PROCESSING COMPLETE

Explanation: This informational message is displayed after SPUFI successfully created the output data set. If errors are encountered before the completion of output data set processing, or during the subsequent BROUSE OUTPUT option for this data set, this message is replaced by either SPUFI or TSO/ISPF error messages.

System action: The system waits for the next user input.

User response: At this point you can change the SPUFI panel variable settings to initiate a new SPUFI request, or you can exit SPUFI.

DSNE363A ONLY RR (REPEATABLE READ) OR CS (CURSOR STABILITY) ALLOWED

Explanation: The user has entered an invalid isolation level. This value must be either RR or CS. 'Isolation level' here refers to the degree to which the SPUFI user wants to isolate SQL execution from the effects of other concurrently executing DB2 applications.

If the repeatable read (RR) level is specified and the AUTOCOMMIT option is 'YES', other applications can modify the selected database values as soon as the SPUFI EXECUTE phase is completed and as soon as the automatic COMMIT (or ROLLBACK, if an error was encountered,) operation is performed. Therefore, it is possible that the retrieved data being browsed by the user has already been modified by another application.

If the RR level is specified and the AUTOCOMMIT option is 'NO', no other DB2 application executing concurrently can change any of the selected database values until the SPUFI user issues either a COMMIT or a ROLLBACK. Therefore, other applications might be in a normal wait state while the SPUFI user is browsing the output data set.

If the cursor stability (CS) level is specified, it does not make any difference whether the AUTOCOMMIT option is 'YES' or 'NO'. Other concurrently executing DB2 applications can change the selected database values as soon as the SELECT has completed.

System action: SPUFI waits until the user specifies either 'RR' or 'CS'.

User response: Type in the correct entry, and press 'ENTER'.

DSNE364A *token* IS INVALID AS AN SQL STATEMENT TERMINATOR

Explanation: The user has entered an SQL statement terminator (token) that is not supported. The following values are not valid for use as a SQL statement terminator:

- blank
- comma
- single and double quotes
- underscore
- left and right parenthesis

System action: The statement cannot be executed.

User response: Specify a valid SQL statement terminator.

DSNE365A SQL STATEMENT TERMINATOR HAS BEEN CHANGED TO *token*

Explanation: The SPUFI input data set contains a #SET statement that changes the SQL statement terminator to *token*.

System action: *token* is the SQL statement terminator until another #SET statement changes it or the end of the input data is reached.

DSNE366A ONLY THE LISTED RECFM TYPES ARE ALLOWED

Explanation: The user has entered an invalid record format for the output data set.

System action: The system waits until the user enters one of the allowed RECFM types.

User response: Enter a RECFM of V, VB, VBA, F, FB, or FBA.

DSNE367A ACTION MUST BE C (COMMIT) OR R (ROLLBACK) OR D (DEFER)

Explanation: This message requests the user to make a commit action decision.

User response: To commit all the changes made since the last commit point, enter a 'C'. To undo all the changes that have been made since the last commit point, enter an 'R'. To defer the decision, enter a blank.

DSNE368A YES OR NO MUST BE ENTERED FOR EACH PROCESSING OPTION

Explanation: One of the processing options was left blank or set to blank by the user.

User response: Enter 'YES' or 'NO' for each processing option.

DSNE370A INPUT DATA SET COULD NOT BE OPENED

Explanation: The input data set could not be opened because of invalid data set attributes. The logical record length (LRECL) of the input data set must be 80K. The record format (RECFM) must be either F or FB. The input data set may be either sequential or partitioned.

User response: Specify an input data set having valid attributes.

DSNE371A OUTPUT DATA SET COULD NOT BE OPENED

Explanation: The output data set could not be opened because of invalid data set attributes. The logical record length (LRECL) must be at least 80KB and less than 32KB. The record format (RECFM) must be F, FB, FBA, V, VB, or VBA. The output data set must be sequential.

User response: Either specify an output data set having valid attributes, or enter correct attributes for this data set.

DSNE372A OUTPUT DATA SET MUST BE SEQUENTIAL

Explanation: The user has requested that a nonsequential output data set contain the output from execution of SQL statements.

User response: Enter the data-set-name of a sequential data set for the output data set.

DSNE374A OUTPUT DATA SET MUST BE SEQUENTIAL

Explanation: The output data set is of unrecognized organization. It must be sequential.

System action: Processing does not begin until the panel entries are correct and the user presses 'ENTER'.

User response: The output data set must be sequential. Enter a sequential data-set-name (no member) in the output data set field.

DSNE375A ACCESS TO INPUT DATA SET NOT AUTHORIZED

Explanation: A local installation-supplied exit has rejected access to the input data set specified by the user.

User response: Either specify the name of a data set for which authorization will be allowed, or arrange to authorize the original data-set-name. Resubmit the request.

DSNE376A ACCESS TO OUTPUT DATA SET NOT AUTHORIZED

Explanation: A local installation-supplied exit has rejected access to the output data set.

User response: Either specify the name of a data set for which authorization will be allowed, or arrange to authorize the original data-set-name. Resubmit the request.

DSNE377A INPUT DATA SET RECFM MUST BE F OR FB WITH LRECL 79 OR 80

Explanation: The user specified an input data set that does not have the required attributes (F or FB with LRECL 79 or 80).

Note: If the input data set has an LRECL of 79, SPUFI assumes the data set was exported by QMF. Therefore, it is recommended that you should not use an LRECL of 79 for a data set that is not exported from QMF.

User response: Correct the input data set attributes, and try again.

DSNE378A ASA CONTROL CHARACTER NOT ALLOWED ON INPUT DATA SET RECFM

Explanation: The user attempted to use an input data set containing forms control characters or machine control characters. These characters are not allowed on the input data set.

User response: Remove the forms control characters or machine control characters from the input data set, and try again.

DSNE379A text from DYNALLOC

Explanation: An attempt to allocate a data set has failed. This message contains the secondary explanation for the failure.

User response: Correct the condition, and resubmit the subcommand.

DSNE381A UNEXPECTED ALLOCATION RETURN CODE return-code, reason-code

Explanation: An unexpected return code from SVC 99 was received. The return code received is the DYNALLOC (SVC 99) return code. Refer to the appropriate MVS publication for explanations of the return and reason codes.

User response: Correct the condition, and resubmit the subcommand.

DSNE383A INVALID OUTPUT DATA SET RECORD FORMAT. OPEN RETURN CODE rcode

Explanation: The data set format was invalid. Acceptable record formats are: F, FB, FBA, V, VB, or VBA.

User response: Enter a valid value in the record format field.

DSNE384A UNEXPECTED DEALLOCATION RETURN CODE return-code, reason-code

Explanation: A deallocation request has returned an unexpected return and reason code.

User response: Refer to the appropriate MVS publication for explanations of the return and reason codes.

DSNE385A ENTER MEMBER NAME. INPUT DATA SET IS PARTITIONED

Explanation: The input data set is partitioned, and the user did not enter the member name on the SPUFI main menu.

System action: Processing does not begin until the panel entries are correct. If any of the processing options are changed, this message is ignored, and processing will proceed according to the new settings.

User response: Enter the member name.

DSNE386A MEMBER member-name NOT PRESENT IN SPECIFIED INPUT DATA SET

Explanation: The named member is not in the input data set.

System action: Processing does not begin until the panel entries are correct.

User response: Enter a correct member name or a correct data-set-name.

DSNE387I DSNTIAR ERROR. RETURN CODE = return-code

Explanation: An internal DB2 error has occurred. DSNTIAR attempted to format a message to explain a nonzero SQL code, but the attempt failed.

User response: Notify the system programmer.

System programmer response: Print the ISPF log, and collect a dump of the TSO address space. Refer to Part 2 of *DB2 Application Programming and SQL Guide* for failure analysis procedures.

DSNE388A INPUT DATA SET MUST BE SEQUENTIAL OR PARTITIONED

Explanation: The input data set organization is not recognized.

System action: Processing does not begin until the panel entries are correct.

User response: Enter a data-set-name with sequential or partitioned organization. It may be necessary to allocate a sequential or partitioned data set.

DSNE389A A MEMBER HAS BEEN SPECIFIED FOR A SEQUENTIAL DATA SET

Explanation: The user has specified a member as input to SPUFI. However, the input data set specified is sequential.

System action: Processing does not begin until the panel entries are correct.

User response: Enter the data-set-name without attempting to specify a member.

DSNE390A THE OUTPUT DATA SET ATTRIBUTES ARE INCONSISTENT

Explanation: This message indicates that the user made one of three possible errors in specifying block size and logical record length (LRECL):

- The user specified a record format (RECFM) of F, but the block size is not equal to the logical record length (LRECL)
- The user specified a record format (RECFM) of F, FB, or FBA, but block size is not evenly divisible by logical record length (LRECL).
- The user specified a record format (RECFM) of VB, but the block size is less than logical record length (LRECL) + 4. The user must allow 4 bytes for the block header.

User response: Specify logical record length (LRECL) and block size appropriately on the DSNESP02 panel.

DSNE391I FREEMAIN ERROR IN MODULE module-name

Explanation: SPUFI encountered a FREEMAIN error in the specified module.

System action: This message is logged in the ISPF log data set. No abend occurs.

User response: Notify the system programmer or system administrator.

System programmer response: Print the ISPF log, and collect a dump of the TSO address space. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNE392I GETMAIN ERROR IN MODULE module-name

Explanation: SPUFI encountered a GETMAIN error in the specified module.

System action: This message is logged in the ISPF log data set. User abend X'00C50003' is issued.

User response: Notify the system programmer or system administrator.

System programmer response: Ensure that enough storage is allocated to the TSO user.

DSNE393I SPF VDELETE ERROR IN MODULE DSNESM90. ERROR CODE = flag

Explanation: An internal SPUFI error has occurred. CSECT DSNESM90 tried to delete an ISPF dialog variable that was not previously defined.

System action: This message is logged in the ISPF log data set. No abend is issued. However, if the problem persists, notify the system programmer.

System programmer response: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: Record the error code 'flag'. It has no user significance, but it can assist the IBM Support Center in resolving the problem.

DSNE395I UNEXPECTED RETURN CODE FROM SPF SERVICE

Explanation: A return code of 4 from ISPF DISPLAY or a return code of 8 from the ISPF editor was encountered.

System action: The message is displayed on the SPUFI panel and logged in the ISPF log data set. No user abend is issued.

User response: Notify the system programmer.

System programmer response: If a new release of ISPF is used, determine whether the above return codes are now assigned meaning by ISPF.

Problem determination: This message may be issued if a previously unassigned (by ISPF) return code from the ISPF DISPLAY or EDIT function is encountered. If a new release of ISPF is used, determine whether the above return codes are now assigned meaning by ISPF.

DSNE396A YOU CANNOT SPECIFY DEFER WHEN LEAVING SPUFI

Explanation: You have attempted to leave SPUFI with outstanding database changes not yet committed or rolled back. You can't leave SPUFI unless a decision has been made to commit or rollback.

System action: The system is waiting for new input to the panel.

User response: Specify COMMIT or ROLLBACK.

DSNE415A *csect-name* **TYPE IS DATE**

Explanation: This is a SPUFI trace message. SPUFI has looked in the SQLTYPE field of the SQLDA and detected a data type of DATE.

DSNE416A *csect-name* **TYPE IS TIME**

Explanation: This is a SPUFI trace message. SPUFI has looked in the SQLTYPE field of the SQLDA and detected a data type of TIME.

DSNE417A *csect-name* **TYPE IS TIMESTAMP**

Explanation: This is a SPUFI trace message. SPUFI has looked in the SQLTYPE field of the SQLDA and detected a data type of TIMESTAMP.

DSNE457I *csect-name* **ARITHMETIC ERROR IN THE CURRENT ROW AND COLUMN**

Explanation: This is a SPUFI trace message. SPUFI has looked in the SQLDA and detected an arithmetic error. The indicator variable associated with this value contains -2.

DSNE601I SQL STATEMENTS ASSUMED TO BE BETWEEN COLUMNS nn AND nn

Explanation: This message indicates which record columns in the input data set are scanned for SQL statements. For data sets of type COBOL, columns 8 through 72 are scanned. For STANDARD data set types, if the LRECL is 79 then columns 1 through 71 are scanned, and if the LRECL is 80 then columns 1 through 72 are scanned.

System action: This message is written to the SPUFI output data set along with other summary messages.

DSNE603E INPUT STATEMENT HAS UNBALANCED APOSTROPHES

Explanation: An opening apostrophe was detected in an input record, but the end of data set was reached before a closing apostrophe was encountered.

System action: SPUFI processing stops. Correct the error, and start again.

User response: A closing apostrophe was probably omitted from a literal value. Add the missing apostrophe, and execute the input file again.

DSNE604E THE SQL STATEMENT EXCEEDS THE MAXIMUM ALLOWED LENGTH

Explanation: This message indicates that the
preceding SQL statement in the output data set exceeds
the maximum length allowed. An SQL statement may
span up to a maximum of 29127 input records.

System action: SPUFI processing stops. Correct the error, and start again.

User response: Edit the SQL statement in question so that it resides on fewer input records.

DSNE605E THE INPUT DATA SET IS EMPTY

Explanation: The user has specified an input data set that does not contain any valid SQL statements.

System action: SPUFI processing stops. Correct the error, and start again.

User response: Edit SQL statements into the specified data set, or specify another input data set that contains valid SQL statements.

DSNE606E INVALID USE OF AN SQL STATEMENT TERMINATOR

Explanation: An SQL statement terminator has been used improperly in an SQL input statement. Either two successive statement terminators were encountered, or a series of blanks followed by a terminator was detected. The proper use of the SQL statement terminator is to delimit the end of a valid SQL statement.

System action: SPUFI processing stops. Correct the error, and start again.

User response: Remove the extraneous terminator from the input data set.

DSNE607E PREVIOUS INPUT RECORD HAS UNBALANCED SO/SI CHARACTERS

Explanation: A shift out (SO) character was detected in the previous input record with no corresponding shift in (SI) character in the record. The SO/SI characters delimit a DBCS string in the SPUFI input data set. SPUFI requires that SO/SI characters be paired within an input record.

System action: SPUFI processing stops.

User response: A closing SI character was probably omitted from a literal value. Add the missing SI character, and re-execute the input file.

DSNE609E INPUT STATEMENT HAS UNBALANCED QUOTES

Explanation: An opening quote (“) was detected in an input record, but a corresponding quote was not found at the end. Two sets of quotes are used to delimit a statement.

System action: SPUFI processing stops.

User response: A closing quote was probably omitted from a literal value. Add the missing quote, and execute the input file again.

DSNE610I NUMBER OF ROWS DISPLAYED IS nn

Explanation: Execution of an SQL SELECT statement causes one or more rows of data to be displayed. This message appears in the output data set following the returned data for a SELECT. It gives a count, 'nn', of the number of rows displayed.

System action: Input data set processing is completed, and this message is included in the SPUFI output.

DSNE611I COLUMN HEADER *name* FOR COLUMN NUMBER *nn* WAS TRUNCATED

Explanation: An SQL SELECT statement was executed, but the specified column name, identified by 'nn' in the message, was truncated. This truncation occurred either because the column name was longer than the remaining record width or because the name was longer than the user-specified maximum field length.

System action: Input data set processing is completed, and this message is included in the SPUFI output.

User response: If the truncation is caused by insufficient record width, use a larger LRECL for the output data set or specify fewer data columns to be returned on the SELECT statement.

If the truncation is caused by the user-specified maximum field length setting, use a larger value for this setting.

DSNE612I DATA FOR COLUMN HEADER name COLUMN NUMBER nn WAS TRUNCATED

Explanation: An SQL SELECT statement was executed, but data for the specified column name, identified by 'nn' in the message, was truncated. This truncation occurred either because the data was longer than the remaining record width or because the data was longer than the user-specified maximum field length.

System action: Input data set processing is completed, and this message is included in the SPUFI output.

User response: If the truncation is because of insufficient record width, use a larger LRECL for the output data set or specify fewer data columns be returned on the SELECT statement.

If the truncation is because of the user-specified maximum field length setting, use a larger value for this setting.

DSNE613I OUTPUT LRECL IS TOO SMALL TO DISPLAY THE FOLLOWING COLUMNS:

Explanation: An SQL SELECT statement was executed, but the logical record length of the output data set was not large enough to display the column names and data values of all returned columns. The columns specified following this message were not displayed because of insufficient remaining record width.

System action: Input data set processing is completed, and this message is included in the SPUFI output.

User response: This condition can be avoided by using a larger LRECL for the output data set or by specifying that fewer data columns be returned on the SELECT statement.

DSNE614I AUTOCOMMIT IS NO, NO CHANGES COMMITTED

Explanation: The user delayed committing or rolling back changes made during execution of the input file until later in the session. This is done by specifying an AUTOCOMMIT option of 'NO'.

System action: Input data set processing is completed, and this message is included in the SPUFI output.

User response: Commit or roll back these changes before ending the session.

DSNE615I NUMBER OF ROWS AFFECTED IS nn

Explanation: This message is written to the output data set following execution of an SQL UPDATE, INSERT, or DELETE statement. 'nn' in the message is a count of the number of rows that were modified as a result of executing the UPDATE, INSERT, or DELETE statement.

Note: In the case of a mass delete of a table in a segmented table space where DB2 does not access nor count the individual rows, this message will not be displayed.

System action: Input data set processing is completed, and this message is included in the SPUFI output.

DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS number

Explanation: This message is written to the output data set following the successful execution of an SQL statement from the input data set. 'number' specifies the SQL return code for the statement. These return codes are listed in *DB2 Codes*.

System action: Input data set processing is completed, and this message is included in the SPUFI output.

DSNE617I COMMIT PERFORMED, SQLCODE IS number

Explanation: When an input file is executed successfully and the user specified an AUTOCOMMIT option of 'YES', this message is written to the output data set to indicate the status of the COMMIT that was performed. 'number' specifies the SQL return code for the COMMIT. These return codes are listed in *DB2 Codes*.

System action: Input data set processing is completed, and this message is included in the SPUFI output.

DSNE618I ROLLBACK PERFORMED, SQLCODE IS number

Explanation: When an error is encountered while executing a SPUFI input file, and the user specified an AUTOCOMMIT option of 'YES', this message will be written to the output data set to indicate the status of the ROLLBACK that was performed. 'number' specifies the SQL return code for the statement. These return codes are listed in *DB2 Codes*.

System action: Input data set processing is completed, and this message is included in the SPUFI output.

DSNE620I NUMBER OF SQL STATEMENTS PROCESSED IS nn

Explanation: This message is written to the output data set after processing all SQL statements in the current input data set. The term 'nn' is a count of the number of SQL statements processed.

System action: Input data set processing is completed, and this message is included in the SPUFI output.

DSNE621I NUMBER OF INPUT RECORDS READ IS nn

Explanation: This message is written to the output data set after processing all SQL statements in the current input data set. The term 'nn' is a count of the total number of records read from the input data set.

System action: Input data set processing is completed, and this message is included in the SPUFI output.

DSNE622I NUMBER OF OUTPUT RECORDS WRITTEN IS nn

Explanation: This message is written to the output data set after processing all SQL statements in the current input data set. The term 'nn' is a count of the total number of records written to the output data set.

System action: Input data set processing is completed, and this message is included in the SPUFI output.

DSNE623I SYNTAX ERROR AT RECORD xx, OFFSET xx

Explanation: A syntax error was detected in the SQL statement preceding this message in the output data set. 'xx' identifies the erroneous line number in the SQL statement, and 'yy' identifies the offset (in characters) of the error within that record.

System action: Input data set processing is completed, and this message is included in the SPUFI output.

User response: Correct the specified syntax error in the input data set, and execute the job again.

DSNE624I UNABLE TO LOAD ISPLINK MODULE. SPUFI PROCESSING TERMINATED

Explanation: During SPUFI initialization processing, an error was encountered attempting to LOAD the ISPF interface module, ISPLINK. This module could not be located in the LNKLST or STEPLIB library concatenation.

System action: SPUFI processing is terminated.

User response: Rearrange your library concatenation so that the ISPLINK module can be located. This module resides in the ISPF load library.

DSNE625I CONNECT TO LOCATION (location name) PERFORMED, SQLCODE IS number

Explanation: The SPUFI user has indicated that the SQL statements defined in the input file are to be processed by the server at the remote location specified in the CONNECT LOCATION field of the SPUFI panel.

This message appears in the output file and indicates that an SQL CONNECT request was issued by SPUFI to the specified location name. The 'number' specifies the SQL return code associated with the CONNECT request. These return codes are listed in *DB2 Codes*.

System action: Subsequent processing of the SQL statements in the input file is based upon the successful completion of the CONNECT request. If the request is unsuccessful, the SQL error number and all associated error messages are placed in the output file.

User response: Respond as indicated by the SQL code returned.

DSNE626I MAXIMUM OUTPUT LINES FOR SELECT STATEMENT REACHED (number), PROCESSING FOR CURRENT SELECT STATEMENT TERMINATED

Explanation: The maximum number of output lines to be displayed for a SELECT statement was reached. Processing of the current SELECT statement is terminated. The maximum number to display is specified by the SPUFI user on the CURRENT SPUFI DEFAULTS panel.

This message appears in the SPUFI output file.

System action: Processing of subsequent SQL statements in the input file continues.

User response: If the number of lines displayed is insufficient, you can increase the maximum number of lines to be displayed for SELECT statements on the CURRENT SPUFI DEFAULTS panel and re-execute the SELECT statement.

DSNE627I *csect-id* **FOR SQLCODE =** *number*, **USER REQUESTED:** *action*

Explanation: During PREPARE processing of the SQL statement, SPUFI received an SQLCODE (+495) and requested you to respond with the desired action to be taken by SPUFI (CONTINUE or BYPASS).

SPUFI does not consider this to be an SQL statement error and places this message in the SPUFI output file to document the SQLCODE received and the action you requested. *number* specifies the SQL return code associated with the PREPARE request. These return codes are listed in *DB2 Codes*.

System action: SPUFI processing for this statement is based on *action* documented in the message:

- CONTINUE
  - Processing for the current SQL statement continues normally.
- BYPASS
  - Processing for the current SQL statement is terminated without error.

User response: Respond as indicated by the SQL code returned.

DSNE800A NO DEFAULT VALUES WERE CHANGED. PRESS ENTER TO CONTINUE

Explanation: The user left panel DSNESP02 without changing it. This panel contains all the defaults that can be modified by the user.

System action: This message is presented on the

SPUFI panel DSNESP01. SPUFI waits for the next user input.

User response: Press 'ENTER' if you want to continue.

DSNE803A INPUT FILE WAS NOT CHANGED. PRESS ENTER TO CONTINUE

Explanation: The user has exited from the ISPF editor without modifying the input data set.

System action: The system waits for the next user input.

User response: Press 'ENTER' if you want to continue. The SPUFI panel variable settings can be changed at this point.

DSNE804A TO CONTINUE, C (COMMIT) OR R (ROLLBACK) MUST BE SPECIFIED

Explanation: The user has specified the AUTOCOMMIT=NO option and now wants either to exit or to switch to another DB2 subsystem with outstanding work uncommitted on the current DB2 subsystem.

System action: DB2 waits for the user to specify either the COMMIT (C) or the ROLLBACK (R) option.

User response: Enter 'C' or 'R' to continue.

DSNE808A EDIT SESSION HAS COMPLETED. PRESS ENTER TO CONTINUE.

Explanation: The user has exited from the ISPF editor.

System action: SPUFI waits form the next user input.

User response: Press 'Enter' if you want to continue. Press 'END' if you want to terminate your SPUFI session. The SPUFI panel variable settings can be changed at this point.

DSNE901I EXECUTION COMPLETE

Explanation: Execution of a subcommand has successfully completed.

System action: The TSO attachment facility is ready for more input.

DSNE903I WARNING, DECLARATION HAS SAME NAME AS TABLE table-name

Explanation: A DCLGEN subcommand has a 'STRUCTURE' parameter that is the same as the 'table name' specified in the message. The requested declaration has been created, but this could cause conflicts with internal names.

System action: DCLGEN execution continues.

User response: Choose a different 'STRUCTURE' parameter.

DSNE904I EXECUTION COMPLETE, MEMBER member-name REPLACED

Explanation: A DCLGEN subcommand executed successfully, replacing the 'member-name' specified in the message.

System action: The TSO attachment facility is ready for more input.

DSNE905I EXECUTION COMPLETE, MEMBER member-name ADDED

Explanation: A DCLGEN subcommand executed successfully, adding the 'member-name' specified in the message to the data set specified in the DCLGEN subcommand.

System action: The TSO attachment facility is ready for more input.

DSNE908E *csect-name* **INVALID USE OF** *parameter* **IN PARAMETER LIST OF KEYWORD** *keyword***.**

Explanation: An invalid parameter was specified in the parameter list of the specified keyword.

Severity: 8

System action: Subcommand processing is terminated. The TSO attachment facility is ready for more input.

User response: Correct the parameter list in error, and resubmit the job.

DSNE909E *csectname* **WARNING, DBCSSYMBOL KEYWORD IS INVALID WITH LANGUAGE** *language* **KEYWORD IGNORED**

Explanation: You specified the DBCSSYMBOL keyword in your DCLGEN invocation, but your LANGUAGE keyword value was not COB2. DBCSSYMBOL only applies to COB2 PICTURE clauses. DCLGEN will ignore this keyword.

System action: DCLGEN processing continues.

User response: In the future, do not specify the DBCSSYMBOL keyword unless the LANGUAGE keyword value is COB2.

DSNE910E *csectname* **WARNING, THE OWNER KEYWORD WAS SPECIFIED WITH A QUALIFIED TABLE NAME. OWNER KEYWORD IGNORED**

Explanation: You specified both the OWNER keyword and a qualified table name for the TABLE keyword in your DCLGEN invocation. The table qualifier will be used as the table owner, and the OWNER keyword will be ignored.

System action: DCLGEN processing continues.

User response: In the future, specify either a qualified table name for the TABLE keyword, or an unqualified table name for the TABLE keyword along with the OWNER keyword.

DSNE920E member-name ALREADY EXISTS, SPECIFY ACTION(REPLACE) TO REPLACE

Explanation: The DCLGEN user tried to overwrite an old declaration with a new one without specifying REPLACE on the 'ACTION' parameter.

System action: The TSO attachment facility is ready for more input.

User response: Change the 'ACTION' parameter to REPLACE (if that's actually what you want to do), and resubmit the command.

DSNE921E REQUIRED PARAMETER table-parameter IS MISSING

Explanation: Your DCLGEN subcommand was missing the required TABLE parameter.

System action: Your DCLGEN subcommand was not processed. The DSN command processor issued a return code of 8. DSN is ready for more input.

User response: Resubmit the DCLGEN subcommand with a TABLE parameter.

DSNE922E TABLE NAME QUALIFIER OF qualifier IS TOO LONG. CANNOT EXCEED 8 CHARACTERS IN LENGTH

Explanation: DCLGEN detected an invalid TABLE parameter. The table name qualifier (the left portion of the table name) was too long.

System action: Your DCLGEN subcommand was not processed. The DSN command processor issued a return code of 8. DSN is ready for more input.

User response: Resubmit the DCLGEN subcommand with a shorter table name qualifier.

DSNE923E CSECT *csect-name* FOUND AN UNRECOGNIZED LANGUAGE *unknown*

Explanation: This is a DCLGEN subsystem error, not a user error. The language parameter 'unknown' is not known to the subsystem.

System action: The TSO attachment facility is ready for more input.

User response: Report the problem to the system programmer.

System programmer response: Re-create the problem with DSN tracing turned on. Collect the trace stream in the DSNTRACE data set. Use the ABEND subcommand of the DSN command processor to collect a dump. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNE924E PARSE FAILED IN CSECT csect-name PARAMETER parm, FLAG flag

Explanation: This is a DB2 subsystem error, not a user error. The 'PARM' parameter was not correctly parsed. The flag specified was detected as an inappropriate value in a parse results field.

System action: The TSO attachment facility is ready for more input.

User response: Report the problem to the system programmer.

System programmer response: Re-create the problem with DSN tracing turned on. Collect the trace stream in the DSNTRACE data set. Use the ABEND subcommand of the DSN command processor to collect a dump. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNE925E BUFFER OVERFLOW IN CSECT csect-name WHILE WRITING string

Explanation: This is a DCLGEN subsystem error. CSECT 'csect-name' detected a buffer overflow condition while attempting to write 'string'.

System action: The TSO attachment facility is ready for more input.

User response: Notify the system programmer.

System programmer response: Re-create the problem with DSN tracing turned on. Collect the trace stream in the DSNTRACE data set. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNE926E COMMAND REJECTED - *command* NOT SUPPORTED UNDER *command_processor*

Explanation: The user entered a TSO TIME or TEST command. These are not supported by DSN or DB2I.

System action: The user should leave DSN or DB2I, return to TSO, and retry the command. DSN can be run under TSO TEST. The TSO attachment facility is ready for more input.

User response: Enter the DSN subcommand, or leave DSN to execute the requested command.

DSNE927E RUN SUBCOMMANDS REQUIRE (1)PROGRAM PARM OR (2)PLAN AND CP PARMS

Explanation: RUN subcommands may contain (1) the 'PROGRAM' parameter and, optionally, the 'PLAN' parameter or (2) the 'PLAN' and command processor ('CP') parameters. Combinations of (1) and (2) are not allowed.

System action: The TSO attachment facility is ready for more input.

User response: Correct the subcommand, and resubmit the job.

DSNE928E RUN SUBCOMMANDS CANNOT HAVE A PROGRAM PARM AND A CP PARM

Explanation: A RUN subcommand cannot contain both the 'PROGRAM' and command processor ('CP') parameters. The subcommand should be resubmitted with only one of the parameters.

System action: The TSO attachment facility is ready for more input.

User response: Correct the subcommand, and resubmit the job.

DSNE929E TABLE NAME LOCATION OF (location-name) IS TOO LONG. CANNOT EXCEED 16 CHARACTERS IN LENGTH.

Explanation: DCLGEN detected an invalid AT(location) keyword parameter. The specified location name is too long.

System action: Your DCLGEN subcommand was not processed. The DSN command processor issues a return code of 8. The DSN is ready for more input.

User response: Resubmit the DCLGEN subcommand specifying a location name that does not exceed 16 characters.

DSNE930E TABLE NAME OF table-name MUST BE QUALIFIED WHEN SPECIFIED WITH LOCATION OF location-name

Explanation: DCLGEN detected an invalid TABLE keyword parameter. You must specify a qualified table name with the AT (location) keyword.

System action: Your DCLGEN subcommand was not processed. The DSN command processor issues a return code of 8. DSN is ready for more input.

User response: Specify a qualified table name and resubmit the DCLGEN command.

DSNE931E *csect-name keyword-name (parameter_value)* **CONTAINS AN INVALID SYMBOL (symbol)**

Explanation: An invalid 'symbol' (symbol) was specified in the 'parameter_value' associated with the specified 'keyword-name'.

Severity: 8

System action: Subcommand processing is terminated. The TSO attachment facility is ready for more input.

User response: Correct the keyword 'parameter_value' in error, and resubmit the job.

DSNE932I WARNING, ONLY IBM SUPPLIED names SHOULD BEGIN WITH DSN

Explanation: The user has issued a BIND, REBIND, or FREE subcommand, using a name or id that begins with DSN.

System action: Processing continues.

User response: In the future, do not use plan-names, collection-ids, or package -ids that begin with DSN. It would be a good idea to FREE the plan or package you just created and to BIND again with a new plan-name, collection-id, or package-id.

DSNE935I ... IS THE DCLGEN COMMAND THAT MADE THE FOLLOWING STATEMENTS

Explanation: This message is a part of normal DCLGEN output. It identifies the DCLGEN command that produced the declaration of which it is a part.

System action: This message is included in the DCLGEN output.

DSNE936I THE NUMBER OF COLUMNS DESCRIBED BY THIS DECLARATION IS nnn

Explanation: This message is a part of normal DCLGEN output. It tells how many columns were in the table that was named in the DCLGEN command.

System action: This message is included in the DCLGEN output.

DSNE937I COBOL DECLARATION FOR TABLE table-name

Explanation: This message is a part of normal DCLGEN output. It tells the name of the table for which the declaration was created.

System action: This message is included in the DCLGEN output.

DSNE938I PLI DECLARATION FOR TABLE table-name

Explanation: This message is a part of normal DCLGEN output. It tells the name of the table for which the declaration was created.

System action: This message is included in the DCLGEN output.

DSNE939E *csect-name* **THE COMBINATION ACQUIRE(ALLOCATE) AND RELEASE(COMMIT) IS NOT ALLOWED**

Explanation: The BIND or REBIND subcommand was submitted with conflicting parameters. You cannot specify ACQUIRE(ALLOCATE) and RELEASE(COMMIT) on the same subcommand.

System action: The subcommand is terminated and the DSN command processor reissues the DSN prompt to allow another subcommand to be entered.

User response: Enter the corrected subcommand. Refer to *DB2 Command Reference* for additional information about the BIND and REBIND subcommands.

DSNE940E DEFER(PREPARE) INVALID WITH NODEFER(PREPARE)

Explanation: The BIND or REBIND subcommand was submitted with conflicting parameters. You cannot specify DEFER(PREPARE) and NODEFER(PREPARE) on the same subcommand because these actions conflict.

System action: The subcommand is terminated and the DSN command processor reissues the DSN prompt to allow another subcommand to be entered.

User response: Enter the corrected subcommand. Refer to *DB2 Command Reference* for additional information regarding the BIND and REBIND subcommands.

DSNE941I DSNBIND IS A RESERVED PLAN NAME

Explanation: This message is generated when the plan name is 'DSNBIND' in a BIND statement.

System action: Subcommand processing is terminated. The TSO attachment facility is ready for more input.

User response: Change the plan-name and resubmit the job.

DSNE942E *csect-name* **REMOTE CONNECTION NAME** *(parameter_value)* **HAS INVALID FORMAT**

Explanation: The REMOTE connection name 'parameter_value' contains a < or > character in the starting or ending position of the parameter, but the parameter value is not bracketed by both characters. This message is also generated if the first character of the connection name is not an alphabetic character.

If the LU name format of the REMOTE keyword is to be used, the name must be specified as <LU_name>.

Severity: 8

System action: Subcommand processing is terminated. DSN command processor reissues the DSN prompt to allow another subcommand to be entered.

User response: Correct the keyword 'parameter_value' in error, and resubmit the job. Refer to *DB2 Command Reference* for additional information about the specification of connection names for the REMOTE keyword.

DSNE943E *csect-name* **REMOTE CONNECTION NAME** *(parameter_value)* **CANNOT EXCEED** *(length_value)* **CHARACTERS IN LENGTH**

Explanation: The REMOTE connection name 'parameter_value' specified is greater than the length 'length_value' permitted for the REMOTE keyword.

This message is issued if the REMOTE location name format exceeds 16 characters, or the LU name format (<LU name>) exceeds 10 characters in length.

Severity: 8

System action: Subcommand processing is terminated. DSN command processor reissues the DSN prompt to allow another subcommand to be entered.

User response: Correct the keyword 'parameter_value' in error and resubmit the job. Refer to *DB2 Command Reference* for additional information about the specification of connection names for the REMOTE keyword.

DSNE944I WARNING, BECAUSE YOUR TABLE OR COLUMN NAMES CONTAIN LOWER CASE OR NON-ALPHABETIC CHARACTERS, YOU WILL NEED TO USE THE SQL (NOT PL/I) INCLUDE STATEMENT TO INCLUDE DCLGEN OUTPUT INTO YOUR PROGRAM

Explanation: DCLGEN had to wrap your table or column name(s) in quotes so DB2 could process them. Quotes are not recognized by the PL/I compiler. To avoid compilation errors, use the SQL INCLUDE statement ('EXEC SQL INCLUDE membername') to merge the DCLGEN output with your source program.

Refer to Chapter 5 of *DB2 SQL Reference* for more information about the SQL INCLUDE statement.

System action: DCLGEN processing continues.

User response: If you have not coded the SQL INCLUDE, do so before the next compile.

DSNE947I QUOTE PARAMETER IS ONLY VALID WITH COBOL. IT HAS BEEN CHANGED TO APOST

Explanation: You specified the DCLGEN QUOTE parameter and a non-COBOL language parameter. This combination is not valid. The QUOTE parameter was specified as either a parameter on the DCLGEN subcommand or a default set for your site at DB2 installation time.

The DCLGEN processor changed the QUOTE setting (for this run) to APOST.

System action: DCLGEN processing continues and should produce valid output.

User response: If you used QUOTE on the DCLGEN command, you may want to change it to APOST before submitting it again.

DSNE948I *csect-id* **FURTHER** *subcommand* **PROCESSING WILL NOT BE ATTEMPTED**

Explanation: A DSN command processor subcommand (indicated in the message) has encountered an error which stopped DSN from processing this subcommand. The subcommand will not complete normally. This message is accompanied by other messages explaining the exact nature of the problem.

System action: Subcommand processing terminates. The DSN prompt is reissued, indicating DSN is ready for another subcommand.

User response: Read the accompanying messages for help in deciding what to do next.

DSNE949E COMMIT PROCESSING ABNORMAL TERMINATION, REASON CODE reason_code

Explanation: The DSN command processor has initiated a 'prepare to commit' request in DB2. The request is rejected when DB2 cannot commit the work performed

System action: Commit processing is terminated.

User response: Examine the reason code associated with this message, and respond as indicated before resubmitting the job.

Problem determination: Reference the associated reason code to determine why the 'prepare to commit' was not successful.

DSNE950E DCLGEN DOES NOT SUPPORT THE lang LANGUAGE

Explanation: A DCLGEN user attempted to specify a LANGUAGE parameter, 'lang', that DCLGEN does not support.

System action: DCLGEN processing terminates.

User response: Correct the LANGUAGE parameter and resubmit the DCLGEN subcommand.

DSNE952I C DECLARATION FOR TABLE table-name

Explanation: This message is a part of normal DCLGEN output. It tells the name of the table for which the declaration was created.

System action: This message is included in the DCGLEN output.

DSNE953I *csect-name* **PRECISION OF DECIMAL COLUMN** *colname* **EXCEEDS THE MAXIMUM PRECISION FOR THE** *lang* **LANGUAGE.**

Explanation: DCLGEN was building a COBOL or PL/I language declaration for a table when it encountered a decimal column with a precision greater than the precision supported by the language.

System action: DCLGEN set the return code to 4 and processed the rest of the table.

User response: This is a warning. Be aware that the declaration for column 'colname' causes a compiler error. You must edit the declaration before including it in an application program.

DSNE954I *csect-name* **USE OF THE DCLGEN COLSUFFIX(YES) OPTION WILL RESULT IN A MIXED FIELD NAME FOR COLUMN** *colname***. THE** *lang* **LANGUAGE DOES NOT SUPPORT MIXED FIELD NAMES.**

Explanation: The COLSUFFIX(YES) option and a double-byte NAMES prefix were specified in a DCLGEN invocation, but column *colname* contains single-byte characters. This combination would result in a generated field name with a mixture of double-byte and single-byte characters, but the *lang* language does not support mixed field names.

System action: DCLGEN set the return code to 4 and generated the field name as requested.

User response: This is a warning. Be aware that the declaration for column *colname* causes a compiler error.

You must edit the declaration before including it in an application program.

DSNE955I *csect-name* **COLSUFFIX(YES) WAS SPECIFIED WITHOUT A NAMES PREFIX. COLUMN NAMES WILL BE USED FOR FIELD NAMES**

Explanation: When COLSUFFIX(YES) is specified in a DCLGEN invocation, DCLGEN generates field names by concatenating the NAMES prefix with the column names. However, no NAMES prefix was specified.

System action: DCLGEN set the return code to 4 and used the column names as field names.

User response: This is a warning. Specify a NAMES prefix when you specify COLSUFFIX(YES).

DSNE956A **A FIELD NAME PREFIX MUST BE SPECIFIED WHEN COLUMN SUFFIX VALUE IS YES.**

Explanation: When YES is specified for the COLUMN SUFFIX field on the DCLGEN panel, DCLGEN generates field names by concatenating the value in the FIELD NAME PREFIX field with the column names. However, no value was specified for FIELD NAME PREFIX.

System action: The DCLGEN panel generated this error message.

User response: Specify a FIELD NAME PREFIX value, or change the COLUMN SUFFIX value to NO.

DSNE960E **PARAMETER STRING EXCEEDS ALLOWABLE LIMIT OF 100 (DECIMAL) CHARACTERS**

Explanation: The PARMS option parameter-string of the RUN subcommand cannot exceed 100 (decimal) characters in length.

System action: The TSO attachment facility is ready for more input.

User response: Correct the PARMS option and resubmit the job.

DSNE964E *csect-name* **INVALID SYNTAX FOR** *keyword* **KEYWORD PARAMETER** *parameter*

Explanation: A parameter that does not follow DB2 naming conventions was specified for *keyword*.

System action: Subcommand processing is terminated. The DSN command processor reissues the DSN prompt to allow you to enter another subcommand.

User response: Correct the keyword parameter value in error, and resubmit the command. Refer to *DB2 Command Reference* for additional information about the specification of a package name and the REBIND/BIND PACKAGE/PLAN or FREE PACKAGE subcommand.

DSNE965E *parameter_value* **IS NOT A VALID** *keyword* **KEYWORD PARAMETER**

Explanation: The *parameter_value* is not a valid parameter value for the keyword.

System action: Subcommand processing terminates. The DSN command processor reissues the DSN prompt to allow you to enter another subcommand.

User response: Enter the corrected subcommand. Refer to *DB2 Command Reference* for additional information about the parameter values supported on the keyword.

DSNE966E **connection_type KEYWORD OMITTED FROM THE ENABLE|DISABLE DECLARATION**

Explanation: The 'connection_type' keyword was not specified in either the ENABLE or DISABLE keywords and is required if connection-names are specified for the 'connection_type'.

Severity: 8

System action: Subcommand processing is terminated. DSN command processor reissues the DSN prompt to allow you to enter another subcommand.

User response: Enter the corrected subcommand. Refer to *DB2 Command Reference* for additional information about the connection type parameter values supported on the ENABLE and DISABLE keywords.

DSNE967E **keyword_name KEYWORD CANNOT BE SPECIFIED MORE THAN ONCE PER RUN**

Explanation: Either the BIND PACKAGE MEMBER or LIBRARY keywords specified more than one parameter value. Only one member or library name is permitted.

Severity: 8

System action: Subcommand processing is terminated. DSN command processor will reissue the DSN prompt to allow another subcommand to be entered.

User response: Enter the corrected subcommand. Refer to *DB2 Command Reference* for additional information regarding the BIND PACKAGE and BIND PLAN subcommands.

DSNE969E *csect-name keyword1* **IS MUTUALLY EXCLUSIVE WITH** *keyword2*

Explanation: Both the PKLIST and NOPKLIST keywords, cannot be specified on one invocation of the REBIND PLAN subcommand. Both the ENABLE and DISABLE keywords cannot be specified on one invocation of the BIND/REBIND PLAN/PACKAGE subcommand.

Severity: 8

System action: Subcommand processing is terminated. DSN command processor will reissue the DSN prompt to allow another subcommand to be entered.

User response: Enter the corrected subcommand. Refer to *DB2 Command Reference* for additional information regarding the REBIND PLAN subcommand.

DSNE970E **EITHER MEMBER OR PKLIST MUST BE SPECIFIED, BOTH ARE ALSO ALLOWED**

Explanation: The BIND PLAN subcommand requires either the MEMBER or PKLIST keyword, and neither was specified. Either keyword or both may be specified.

Severity: 8

System action: Subcommand processing is terminated. DSN command processor will reissue the DSN prompt to allow another subcommand to be entered.

User response: Enter the corrected subcommand. Refer to *DB2 Command Reference* for additional information regarding the BIND PLAN subcommand.

DSNE971E **EITHER MEMBER OR COPY BUT NOT BOTH MUST BE SPECIFIED**

Explanation: The BIND PACKAGE subcommand requires either the MEMBER or COPY keyword; both are not allowed on the same subcommand invocation.

Severity: 8

System action: Subcommand processing is terminated and the DSN command processor will reissue the DSN prompt to allow another subcommand to be entered.

User response: Enter the corrected subcommand. Refer to *DB2 Command Reference* for additional information regarding the BIND PACKAGE subcommand.

DSNE974E *csect-name keyword1* **KEYWORD IS INVALID WITH** *keyword2* **KEYWORD.**

Explanation: The keyword2 should not be specified with keyword1.

System action: Subcommand processing is terminated. The DSN command processor reissues the DSN prompt to allow another subcommand to be entered.

User response: Modify one of the keywords on the command and resubmit the job. Refer to *DB2 Command Reference* for additional information about the BIND PACKAGE/PLAN subcommands.

DSNE975E *csect-name keyword* **KEYWORD IS ONLY VALID WHEN COPY KEYWORD IS PRESENT**

Explanation: A keyword was issued that may only be specified in conjunction with the COPY keyword.

keyword
COPYVER or OPTIONS

Severity: 8

System action: Subcommand processing is terminated. The DSN command processor will reissue the DSN prompt to allow another subcommand to be entered.

User response: Enter the corrected subcommand. Refer to *DB2 Command Reference* for additional information regarding the BIND PACKAGE subcommand.

DSNE976E **LIBRARY KEYWORD IS ONLY VALID WHEN MEMBER KEYWORD IS PRESENT**

Explanation: The LIBRARY keyword may only be specified in conjunction with the MEMBER keyword.

Severity: 8

System action: Subcommand processing is terminated. The DSN command processor will reissue the DSN prompt to allow another subcommand to be entered.

User response: Enter the corrected subcommand. Refer to *DB2 Command Reference* for additional information regarding the BIND PACKAGE subcommand.

DSNE999 **INVALID SQL RETURN CODE**

Explanation: The number you selected is not a valid SQL return code.

This message is issued by panel: DSN4SRC

User response: Check the number and enter it again.

Chapter 5. DSNF... messages

DSNF001I MESSAGE GENERATOR INITIALIZATION PARAMETERS NOT FOUND. DEFAULTS ASSUMED

Explanation: The message generator was unable to access the routing code initialization parameter defined by the DSN6SYSP macro. Default values defined by that macro are assumed.

System action: Subsystem initialization continues.

Operator response: Notify the system programmer.

System programmer response: It might be necessary to change the DSN6SYSP macro and replace the DSN6SYSP member of the DSNIPARM directory. Refer to Part 2 of *DB2 Installation Guide* for details of this process.

DSNF700I *csect-name message-id* **HAS BEEN ISSUED**

Explanation: An unsuccessful attempt has been made to issue the message 'message-id'. This message is issued only if the requested message could not be found in the DB2 message directory.

System action: The system proceeds as though the requested message had been issued.

Operator response: Notify the system programmer.

System programmer response: Refer to this manual for actual message text. You need the message number ('message-id'). If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

© Copyright IBM Corp. 1983, 2008

Chapter 6. DSNG... messages

DSNG001I EDM *stype* CAN NOT BE DECREASED BELOW THE VALUE *isize* SPECIFIED AT DB2 STARTUP

Explanation: The EDM *stype* storage cannot be decreased below the size that was specified at DB2 startup.

System action: Processing continues.

DSNG002I EDM *stype* HAS AN INITIAL SIZE *isize*, REQUESTED SIZE *rsize*, AND AN ALLOCATED SIZE *asize*

Explanation: This message is issued in response to a request to increase or decrease the EDM *stype* storage.

isize The EDM *stype* storage size prior to the request for a change.

rsize The requested EDM *stype* storage size.

asize The EDM *stype* storage size immediately available.

- When increasing the EDM Pool, the *asize* is the storage available to satisfy the request.
- When decreasing the EDM Pool, the *asize* is the INITIAL size reduced by the amount that could be released immediately.

  When the *asize* is larger than the *rsize*, the difference is marked to be released when it is no longer referenced.

System action: Processing continues.

DSNG003I INSUFFICIENT VIRTUAL STORAGE TO EXPAND EDM *stype* STORAGE

Explanation: Insufficient virtual storage exists to expand the EDM *stype* storage as specified.

System action: The EDM *stype* storage is expanded only to the size allowed by the available virtual storage.

DSNG004I EDM *stype* HAS A FINAL ALLOCATED SIZE *asize*

Explanation: This message is issued in response to a request to decrease the EDM *stype* storage. The message is issued when the requested new level is reached.

stype The EDM pool that is being reduced. Possible values are Pool, DBD Pool, or Satement Pool.

asize The EDM *stype* storage that is available.

System action: Processing continues.

DSNG005I DBD IS INCONSISTENT WITH THE CATALOG. UNABLE TO FIND A ROW IN SYSTABLES WITH DBID *dbid* AND OBID *obid*.

Explanation: This message is issued in response to a request to load a database with DBID *dbid*. This message is issued when the row is not found in SYSTABLES.

dbid DBID of the database that is being loaded.

obid OBID of a table in the database.

Severity: 8 (error).

System action: DBD could not be loaded.

Operator response: Based on *dbid* and *obid* from the message, verify that the row does not exist in the Catalog. REPAIR DBD TEST or DIAGNOSE can be run to gather more diagnostic information. Once you have determined that the Catalog contains the correct information, REPAIR DBD REBUILD can be run to rebuild the DBD.

DSNG006I DBD IS INCONSISTENT WITH THE CATALOG. UNABLE TO FIND A ROW IN SYSCOLUMNS WITH DBID *dbid*, OBID *obid*, AND COLUMN NUMBER *colno*.

Explanation: This message is issued in response to a request to load a database with DBID *dbid*. This message is issued when the row is not found in SYSCOLUMNS.

dbid DBID of the database that is being loaded.

obid OBID of a table in the database.

colno Column number in the table.

Severity: 8 (error).

System action: DBD could not be loaded.

Operator response: Based on *dbid/obid/colno* from the message, verify that the row does not exist in the Catalog. REPAIR DBD TEST or DIAGNOSE can be run to gather more diagnostic information. Once you have determined that the Catalog contains the correct information, REPAIR DBD REBUILD can be run to rebuild the DBD.

DSNG007I DB2 CATALOG LEVEL (*cat*) CODE LEVEL (*code*) MODE (*mode*)

Explanation: This message is issued when the catalog is loaded. It displays DB2 catalog release level, code release level and the mode indicator.

© Copyright IBM Corp. 1983, 2008

cat DB2 catalog release level. For example, 810 for V8.

code DB2 code release level. For example, 810 for V8.

mode C for compatibility mode, E for enabling-new-function mode, and N for new-function mode.

System action: Processing continues.

Chapter 7. DSNH... messages

These messages describe messages from the precompiler and the compiler. This section contains the following topics:

- "Message line numbers for DSNH... messages"
- "Message severity codes for DSNH... messages"
- "Source tokens in DSNH... messages" on page 102
- "Messages in DSNH" on page 102

Message line numbers for DSNH... messages

The line number *nnnn* in the compiler messages is generated by assigning sequential numbers to the original source. The numbers are printed on the SQL precompiler source listing (if the SOURCE option is specified to the precompiler).

'COL *cc*' in the messages identifies the source column in which the condition was detected. The inidcated column, '*cc*', usually specifies the location of the column containing the first character of the associated source token. However, if the precompiler cannot determine this location, it is the beginning of the statement. Because the detected error might be similar to a valid, but different, construction, the SQL parsing might indicate the wrong location for the error.

Message severity codes for DSNH... messages

One of the characters I, W, E, S, or U is shown at the beginning of each precompiler message as an aid to the programmer. These characters indicate the severity level as follows:

I	0	Informational
W	4	Warning
E	8	Error
S	12	Severe error
U	16	Unrecoverable error

When unsupported non-DB2 statements are detected during SQL(ALL) precompilation, the return code will be forced to 4 if the return code was 8, and no host-related errors were detected. This is done so that the application can proceed with the compilation and link-edit steps. This allows the syntax checking to be delayed until bind. If a severity 8 precompiler error message is not a host-related error, the precompiler treats the error as a warning and place a W instead of an E in the precompiler output listing. In this instance, the precompiler also issues DSNH049I to indicate that the actual return code 8 from the precompiler was changed to 4.

The following precompiler messages with severity levels of 8 are not changed from errors to warnings when SQL(ALL) is in effect:

DSNH051I
DSNH080I
DSNH290I
DSNH291I

© Copyright IBM Corp. 1983, 2008

DSNH292I
DSNH312I
DSNH313I
DSNH314I
DSNH504I

Source tokens in DSNH... messages

Tokens from your source program do not always appear in messages exactly as they appear in your program. For example, leading zeros are removed from integer tokens.

Messages in DSNH

DSNH003I S *csectname* **FILE** *ddname* **PARAMETERS INCORRECT**

Explanation: The file *ddname* has invalid or incorrect parameters, such as logical record length.

Severity: 12 (severe error)

System action: The precompiler is terminated.

User response: Correct the data set or library parameters.

DSNH004I S *csectname type* **COULD NOT BE FOUND**

Explanation: The precompiler could not find the indicated type of statement in the application program source. This message is only issued for HOST(COBOL) and HOST(COB2). The *type* is DATA DIVISION, WORKING-STORAGE SECTION, or PROCEDURE DIVISION. This error can cause other errors.

Severity: 12 (severe error)

System action: Processing continues. Additional errors might result.

User response: Ensure that the host language statement in your application program is correct and that the statement is present within the proper margins.

DSNH005I U *csectname* **FILE** *ddname* **NO MEMBER NAME SPECIFIED**

Explanation: The *ddname* must refer to a partitioned data set, and the member name must be specified in the allocation. This helps to prevent accidental erasure of the DBRM library.

Severity: 16 (unrecoverable error)

System action: The precompiler is terminated.

User response: Correct the JCL or TSO allocate command to specify a member name.

DSNH006I U *csectname* **FILE** *ddname* **COULD NOT BE OPENED**

Explanation: The file identified in the message is required, but it could not be opened.

Severity: 16 (unrecoverable error)

System action: The precompiler is terminated.

User response: Ensure that the file is present in your JCL.

System programmer response: Ensure that the SYSIN data is provided. Also ensure that SYSLIB is defined if the application program contains SQL INCLUDE statements.

DSNH007I E *csectname* **LINE** *nnnn* **COL** *cc* **ILLEGAL CHARACTER IGNORED: '***c***' ('***hh***'X)**

Explanation: The character identified in the message as *c* is not permitted. The hexadecimal representation for the character is *hh*.

Severity: 8 (error)

System action: The character is ignored.

User response: Correct the character. You might need
to place the character within quotation marks or
apostrophes. Check to ensure that the CCSID in which
the source is written matches the CCSID option for the
Precompiler. For example, if the CCSID 1047 codepoints
for left and right square brackets (hexadecimal 'AD'
and 'BD') are used in a C source program, ensure that
the Precompiler CCSID option is CCSID(1047).

DSNH008I S *csectname* **LINE** *nnnn* **COL** *cc* **NESTED INCLUDE NOT SUPPORTED**

Explanation: While processing an INCLUDE, another nested INCLUDE statement was encountered.

Severity: 12 (severe error)

System action: The nested INCLUDE statement is ignored.

User response: Remove the nested INCLUDE statements; they are not allowed in an included file. The DB2 precompiler permits only one level of INCLUDE statements.

DSNH009I **S** *csectname* **LINE** *nnnn* **COL** *cc* **MEMBER** *name* **COULD NOT BE INCLUDED**

Explanation: The identified member name was not found or could not be read.

Severity: 12 (severe error)

System action: Processing continues without the data from that member.

User response: Ensure that the member is in the library and that its name is spelled correctly. Check and correct any JCL errors. They are likely to cause other errors. Ensure that the data set and library parameters are correct.

DSNH010I **S** *csectname* **LINE** *nnnn* **COL** *cc* **UNTERMINATED STRING CONSTANT BEGINNING** *text*

Explanation: The constant identified in the message as '*text*' is not terminated properly.

Severity: 12 (severe error)

System action: The string is treated as if it were terminated by the end of file.

User response: Terminate the string constant. Check for and correct any missing or extra quotation marks and apostrophes. These errors are likely to cause others. Some statements were probably missed because of missing or extra string delimiters.

DSNH011I **S** *csectname* **LINE** *nnnn* **COL** *cc* **UNTERMINATED COMMENT**

Explanation: A comment is not terminated properly. This message is only issued for HOST(PLI) and HOST(C).

Severity: 12 (severe error)

System action: The string is treated as if it were terminated by the end of file.

User response: Ensure that the comment terminator is typed properly and that it is within the specified margins. This error is likely to cause others. Some statements were probably missed because of the missing comment terminator.

DSNH012I **W** *csectname* **LINE** *nnnn* **COL** *cc* **INVALID CONTINUATION**

Explanation: The continuation does not follow the host language rules.

Severity: 4 (warning)

System action: Processing continues. Additional errors might result.

User response: Follow the host language continuation rules and embedded SQL continuation rules. Check the margin option. This error is most common in assembler language in cases where a continuation begins before the continuation column (the continuation column is column 16 by default).

DSNH013I **E** *csectname* **LINE** *nnnn* **COL** *cc* **SQL STATEMENT FOUND OTHER THAN IN THE FIRST PROGRAM (OR MAIN PROGRAM)**

Explanation: This message is for VS COBOL II only. If any SQL statement is found in the source after the first program in a BATCH COMPILE or the end of the main program of a nested program, this message is issued.

Severity: 8 (error)

System action: The precompiler is terminated.

User response: Move up the SQL statement and corresponding host variable declaration to the main program or to the first program and resubmit the job.

DSNH014I **S** *csectname* **LINE** *nnnn* **COL** *cc* **SQL STATEMENT IS FOUND IN AN INAPPROPRIATE PLACE**

Explanation: If any SQL statement is found in the COBOL DECLARATIVES section, this message is issued.

Severity: 12 (severe error)

System action: Processing continues. Additional errors might result.

User response: Move the SQL statement out of the COBOL DECLARATIVES section and resubmit the job.

DSNH016I **E** *csectname* **LINE** *nnnn* **COL** *cc token* **REQUIRED**

Explanation: The required character or keyword identified as *token* is missing.

Severity: 8 (error)

System action: The token is assumed to be present in order to continue checking the statement syntax.

User response: Correct the SQL statement. Determine if you skipped either this keyword or another keyword earlier in the SQL statement. The assumed token is printed in the SYSTEM error message. This token might not be what you intended, even though it is the only symbol allowed in the clause at this point.

DSNH019I **W** *csectname* **OPTION: INVALID OPTION** *option*

Explanation: The identified option was requested, but it is not a precompiler option.

Severity: 4 (warning)

System action: The option is ignored.

User response: Check the valid precompiler options to determine if your spelling is correct for this option. This problem might cause additional errors if the APOST, APOSTSQL, COMMA, HOST, MARGINS, QUOTESQL, or TWOPASS option is needed but not specified. The other options are either defaults or affect only the listing. Correct the problem and run the precompile again.

DSNH021I **W** *csectname* **OPTION: SUBOPTION NOT PERMITTED WITH OPTION** *option*

Explanation: The identified option cannot have suboptions, but one or more were specified.

Severity: 4 (warning)

System action: The suboptions are ignored.

User response: None is required. However, you should remove the suboption.

DSNH022I **W** *csectname* **OPTION: UNMATCHED ')' TAKEN AS ','**

Explanation: An unmatched ending parenthesis was found before the end of the option string.

Severity: 4 (warning)

System action: The parenthesis is assumed to be a comma.

User response: Balance the parentheses.

DSNH023I **W** *csectname* **OPTION: SUBOPTION MISSING FOR OPTION** *option*

Explanation: The specified option requires a suboption, but one was not provided.

Severity: 4 (warning)

System action: The option is ignored.

User response: Provide a suboption or remove the option. This problem might cause additional errors if the HOST or MARGINS option is needed but not specified. The other options are either defaults or affect only the listing. Correct the problem and run the precompile again.

DSNH024I **W** *csectname* **OPTION: SUBOPTION** *suboption* **INVALID FOR OPTION** *option*

Explanation: The identified suboption is not allowed for the option specified.

Severity: 4 (warning)

System action: The entire option is usually ignored.

User response: Refer to the 'options used' list in the precompiler output listing to see if the option was used. Remove the suboption.

DSNH025I **W** *csectname* **OPTION: EXCESSIVE SUBOPTIONS FOR OPTION** *option*

Explanation: Too many suboptions were given for the identified option.

Severity: 4 (warning)

System action: The excess suboptions are ignored.

User response: Remove excess suboptions.

DSNH026I **W** *csectname* **OPTION: INVALID MARGINS FOR HOST LANGUAGE**

Explanation: The host language does not permit the specified margins.

Severity: 4 (warning)

System action: The MARGINS option is ignored.

User response: Ensure that the source statements are within the proper margins and that the margins are specified correctly. The margins for COBOL must be 8, 72. The margins for FORTRAN must be 1, 72.

DSNH027I **W** *csectname* **SUBOPTION** *suboption* **TOO LONG, TRUNCATED AFTER** *numofchar* **CHARS**

Explanation: The version ID is greater than *numofchar* bytes long. The maximum length for the version ID is 64 bytes.

Severity: 4 (warning)

System action: The option is honored, but the version ID is truncated after 64 bytes.

User response: Refer to the 'options used' list in the precompiler output to see what version ID was used.

DSNH028I **W** *csectname* **ATTACH SUBOPTION** *suboption* **NOT PERMITTED WITH HOST (FORTRAN) SPECIFIED.**

Explanation: The identified suboption for precompiler option ATTACH cannot take effect when the application is a FORTRAN application because the interface code for FORTRAN is always DSNHFT.

Severity: 4 (warning)

System action: The option is ignored.

User response: None is required. However, you can either code ATTACH(TSO) or specify no ATTACH precompiler option

---

DSNH029I **E** *csectname* **LINE** *nnnn* **COL** *cc* **INTO CLAUSE REQUIRED**

Explanation: SELECT statements embedded in an application program must have an INTO clause to denote where the results of the SELECT are to be placed. Dynamic SELECT statements do not permit the INTO clause.

Severity: 8 (error)

System action: The statement is replaced by host language statements that, if executed, set an error code in the SQLCA.

User response: Add the INTO clause to the SELECT statement, and precompile the application program again.

DSNH030I **W** *csectname* **LINE** *nnnn* **COL** *cc* **STATEMENT CONTAINS WRONG NUMBER OF 'INTO' VARIABLES**

Explanation: The number of variables specified in an INTO clause is not the same as the number of select-list elements (taking into account expansion of host structures and '*' or 'table.*' select entries according to the declarations supplied).

Severity: 4 (warning)

System action: Processing continues.

User response: Ensure the included table declarations are correct. If the declarations are correct, this statement causes an error when the application program is bound. If the declarations are incorrect, you can proceed, but the declarations should be corrected. Otherwise, you need to correct the SQL statement and precompile again.

DSNH031I **W** *csectname* **LINE** *nnnn* **COL** *cc* **STATEMENT CREATES A COLUMN WITH NAME** *name1* **WHICH DOES NOT MATCH DECLARED NAME** *name2*

Explanation: The column name specified in a CREATE statement (*name1*) is not the same as the name of the corresponding column in the DECLARE TABLE statement (*name2*).

Severity: 4 (warning)

System action: Processing continues.

User response: Ensure that the DECLARE TABLE matches the CREATE TABLE. You can use the cross-reference and source listings to help find the two statements.

DSNH032I **W** *csectname* **LINE** *nnnn* **COL** *cc* **STATEMENT CREATES A COLUMN** *name* **WHOSE ATTRIBUTES DO NOT MATCH THE DECLARATION**

Explanation: The column *name* is specified in a CREATE statement with a different data type or length than was specified for the corresponding column in the DECLARE TABLE statement.

Severity: 4 (warning)

System action: Processing continues.

User response: Correct the DECLARE TABLE or CREATE TABLE statement. If the DECLARE TABLE statement is incorrect, some warnings might also be incorrect or might not be generated when needed. If the CREATE TABLE statement is incorrect, you must correct it.

DSNH033I **W** *csectname* **LINE** *nnnn* **COL** *cc* **STATEMENT CREATES MORE OR LESS COLUMNS THAN WERE DECLARED**

Explanation: The number of columns specified in a CREATE statement is different from the number of columns specified in the DECLARE TABLE statement.

Severity: 4 (warning)

System action: Processing continues.

User response: Ensure that the DECLARE TABLE matches the CREATE TABLE. If the DECLARE TABLE statement is incorrect, some warnings might also be incorrect or might not be generated when needed. If the CREATE TABLE statement is incorrect, you must correct it.

DSNH034I **W** *csectname* **LINE** *nnnn* **COL** *cc* **STATEMENT INSERTS MORE OR LESS VALUES THAN THE NUMBER OF COLUMNS DECLARED**

Explanation: The number of values specified in an INSERT statement is different from the number of columns specified in the DECLARE TABLE statement for the receiving table.

Severity: 4 (warning)

System action: Processing continues.

User response: Insert the correct number of values. If the DECLARE TABLE statement is incorrect, this warning might also be incorrect. If the INSERT statement is incorrect, you must correct it.

DSNH038I **S** *csectname* **LINE** *nnnn* **COL** *cc* **PROGRAM OR SUBPROGRAM CONTAINS MORE THAN 32767 SQL STATEMENTS**

Explanation: The number of SQL statements in an application program or a FORTRAN subprogram is greater than 32767.

Severity: 12 (severe error)

System action: The precompiler terminates.

User response: Divide the application program or subprogram into smaller programs or reduce the SQL statements in each program unit so that the maximum number of SQL statements is not exceeded.

DSNH041I **W** *csectname* **OPTION: MISSING HOST OPTION**

Explanation: The host language was not specified in the OPTION field.

Severity: 4 (warning)

System action: HOST(COBOL) is assumed.

User response: Specify the host language if it is not COBOL. It is probably best to specify the host language and remove all warnings.

DSNH042I **W** *csectname* **OPTION:** *option* **OPTION INVALID FOR HOST LANGUAGE**

Explanation: The option identified in the message is not valid. The QUOTE and COMMA options are not permitted with HOST(ASM), HOST(PLI), or HOST(C). The TWOPASS option is not permitted with HOST(COBOL).

Severity: 4 (warning)

System action: The specified option is ignored.

User response: Remove the invalid option, or correct the host specification.

DSNH049I **W** *csectname* **RETURN CODE IS FORCED TO 4 BECAUSE OPTION SQL(ALL) IS IN EFFECT, ORIGINAL RETURN CODE WAS** *nnn*

Explanation: One or more SQL-related error messages were detected by the precompiler, but because of option SQL(ALL), the return code is forced to 4 in order to let the application proceed with compilation.

The SQL syntax checking is deferred until bind time because the source might contain non-DB2 SQL syntax that the DB2 precompiler does not recognize and flags as an error. The DBRM produced by this precompilation cannot be bound successfully by DB2 if there are SQL syntax errors or non-DB2 SQL syntax.

The return code remains untouched if there are host-related syntax or semantic errors in the source.

Severity: 4 (warning)

System action: Processing continues.

User response: Ensure that the intention of this precompilation is for executing it in a non-DB2 system.

DSNH050I **I** *csectname* **WARNINGS HAVE BEEN SUPPRESSED DUE TO LACK OF TABLE DECLARATIONS**

Explanation: One or more warning messages were suppressed because no table declarations were found before the first SQL statement that referenced a table or column.

Severity: 0

System action: Processing continues. Suppression is discontinued if a table declaration is found.

User response: Insert (or INCLUDE) table declarations before they are referenced in order to permit table and column checking.

DSNH051I **E** *csectname* **LINE** *nnnn* **COL** *cc sqltype name* **PREVIOUSLY DECLARED OR REFERENCED**

Explanation: The variable *sqltype* is TABLE, CURSOR, or STATEMENT. The identifier *name* was already defined with the given *sqltype*. The definition might have occurred implicitly. For example, *name* might have been found in the FROM clause of a SELECT, causing it to be defined as a table.

Severity: 8 (error)

System action: The name is installed in the symbol table as a duplicate.

User response: Remove duplicate definitions, or correct the references. Avoid implicit definitions by inserting (or including) table declarations before they are referenced. Check the number of passes used. Use the cross-reference and source listings to help resolve this type of error.

DSNH052I **E** *csectname* **LINE** *nnnn* **COL** *cc* **COLUMN** *name* **IS ALREADY DEFINED IN TABLE** *table-name*

Explanation: The identified name is already defined as a column in the table named *table-name*. The definition might have occurred implicitly. For example, *table-name* might have been found in a select-list, causing *name* to be defined as a column within *table-name* if *table-name* was not declared as a host variable.

Severity: 8 (error)

System action: The name is installed in the symbol table as a duplicate.

User response: Remove duplicate definitions, or correct the references. Avoid implicit definitions by inserting (or including) table declarations before they are referenced.

DSNH053I **W** *csectname* **NO SQL STATEMENTS WERE FOUND**

Explanation: No SQL statements were found in the precompiler input file. An empty DBRM member was written.

Severity: 4 (warning)

System action: Processing continues.

User response: If you embedded SQL statements in your application program and none were found, make sure your column margins are properly defined for your application program.

DSNH060I **E** *csectname* **LINE** *nnnn* **COL** *cc* **INVALID** *type* **SPECIFICATION:** *spec*

Explanation: The *type* is either SCALE or LENGTH. The *spec* is the given SCALE or LENGTH specification in an SQL declaration. Length and scale specifications must be unsigned integers. The scale specification, where permitted, cannot exceed the length. The length specification cannot be 0.

Severity: 8 (error)

System action: Processing continues.

User response: Correct the length or scaling.

DSNH080I **E** *csectname* **LINE** *nnnn* **COL** *cc usage* **VARIABLE** *name* **IS NOT** *type* **TYPE**

Explanation: The *usage* is INDICATOR or STRING. The *type* is SMALLINT, VARCHAR, or CHARACTER. The data type of the variable *name* is not acceptable. All indicator variables must be small integers. PREPARE and EXECUTE strings must be variable character strings, except for FORTRAN, which requires character type.

Severity: 8 (error)

System action: If the containing SQL statement is a declaration, it is discarded. If the containing SQL statement is executable, it is replaced by host language statements that will, if executed, set an error code in the SQLCA.

User response: Specify a host variable of the proper type.

DSNH081I **W** *csectname* **LINE** *nnnn* **COL** *cc* **STATEMENT DOES NOT CONFORM TO ANS SYNTAX STANDARD - REASON CODE** *ccc*

Explanation: When the precompiler option STDSQL(YES) is in effect, the SQL syntax is checked to see if it conforms to the ANSI/ISO SQL standard of 1992. Reason codes are:

000 Aggregate function with DISTINCT as an arithmetic-operation operand does not conform to the ANSI/ISO SQL standard of 1992.

001 FOR UPDATE OF as part of a DECLARE CURSOR statement does not conform to the ANSI/ISO SQL standard of 1992.

Severity: 4 (warning)

System action: Processing continues.

User response: You have the option of changing or not changing the syntax to conform to the ANS syntax standard.

System programmer response: Check the SQL statement.

DSNH082I **I** *csectname* **LINE** *nnnn* **COL** *cc* **STATEMENT DOES NOT CONFORM TO IBM SQL SYNTAX OR SEMANTIC STANDARD - REASON CODE** *ccc*

Explanation: When the precompiler option SQL(ALL) is in effect, statements that do not conform to certain IBM SQL rules are flagged.

Severity: 0 (informational)

System action: Processing continues.

User response: Change the syntax of the statement to conform to the IBM SQL standard if portability is required. Not changing the syntax means that the program is less portable within the IBM SQL environment.

DSNH083I **I** *csectname* **LINE** *nnnn* **COL** *cc* **IBM SQL RESERVED WORD** *ibmkeywd* **SPECIFIED AS AN IDENTIFIER NAME**

Explanation: When the precompiler option SQL(ALL) is in effect, IBM SQL reserved words should not be used as identifier names. Using IBM SQL reserved words limits the portability of the statement.

Severity: 0 (informational)

System action: Processing continues.

User response: If possible, change the identifier name to a non-IBM SQL reserved word so that the statement conforms to the IBM SQL standard. Not changing the identifier name means that the program is less portable within the IBM SQL environment.

DSNH084I W *csectname* **LINE** *nnnn* **COL** *cc* **UNACCEPTABLE SQL STATEMENT**

Explanation: The referenced SQL statement is not allowed in this situation, but it will not cause an error. Depending on the option specified at precompile time, one of the following occurred.

- The BEGIN DECLARE SECTION(*), END DECLARE SECTION(*), and UPDATE STATISTICS statements are flagged as not compatible.

  * For C applications, BEGIN and END DECLARE SECTION are not flagged.

  * For other languages, the statement is only allowed with precompiler option STDSQL(YES).
- Using the STDSQL(YES) precompiler option, the INCLUDE SQLCA statement is flagged as a warning.

Severity: 4 (warning)

System action: The BEGIN DECLARE SECTION, END DECLARE SECTION, and INCLUDE SQLCA statements are ignored. The other statements are replaced by host language statements that will, if executed, set a warning code or SQLCODE -84 in the SQLCA.

User response: Unless this application program is used in a portable manner, remove the statement.

DSNH088I W *csectname* **LINE** *nnnn* **COL** *cc* **THIS STATEMENT WILL** *action* **AN ENTIRE TABLE**

Explanation: The *action* is either DELETE or UPDATE. The referenced SQL statement has no WHERE clause and will delete or update all rows of the specified table. This message is printed as a warning to the programmer.

Severity: 4 (warning)

System action: Processing continues.

User response: Be sure that you intend to delete or update all rows, or add a suitable WHERE clause.

DSNH090I S *csectname* **LINE** *nnnn* **COL** *cc* **HOST VARIABLE NOT PERMITTED HERE:** *name*

Explanation: Host variable references are not permitted in this type of SQL statement. The *name* is invalid.

Severity: 12 (severe error)

System action: The statement is replaced by host language statements that will, if executed, set an error code in the SQLCA.

User response: Specify a literal or a column name rather than a host variable. The colon (:) indicates that the name that follows is a host variable. Remove it to specify a column name.

DSNH091I I *csectname* **LINE** *nnnn* **COL** *cc* **EBCDIC CCSID used for string-expression.**

Explanation: In a PL/1 application program, a string-expression was specified for a PREPARE or EXECUTE IMMEDIATE statement, and the precompiler-generated structures for this string-expression specify an EBCDIC CCSID.

Severity: 0 (informational)

System action: Processing continues.

User response: If you intended that the string-expression should be treated as a string expression, then no action is required.

If the string-expression consists of just one host variable, and you intended that the string-expression should be treated as a host variable, then the host variable must be preceded by a colon, and at least one DECLARE VARIABLE statement must be contained in your program source.

If you add a DECLARE VARIABLE for any host variable in your program source, and re-precompile your application, the variable previously treated as a string-expression by the precompiler will be treated as a host variable and this message will not be issued.

DSNH097I E *csectname* **LINE** *nnnn* **COL** *cc***. THE USE OF LONG VARCHAR OR LONG VARGRAPHIC IS NOT ALLOWED IN THIS CONTEXT.**

Explanation: The statement attempted to use the LONG VARCHAR or LONG VARGRAPHIC syntax. This syntax cannot be used for the following statements:

- CAST syntax
- CREATE DISTINCT TYPE
- CREATE FUNCTION
- CREATE PROCEDURE
- ALTER FUNCTION
- COMMENT ON FUNCTION
- GRANT EXECUTE ON FUNCTION
- REVOKE EXECUTE ON FUNCTION

Severity: 8 (error)

System action: The statement cannot be executed.

User response: Correct and reissue the failing statement.

DSNH101I S *csectname* **THE STATEMENT IS TOO LONG OR TOO COMPLEX.**

Explanation: The statement cannot be executed because it exceeds the system limits for either length or complexity.

Severity: 16

System action: The statement cannot be executed.

User response: Break the statement into shorter or less complex SQL statements.

DSNH102I **E** *csectname* **LINE** *nnnn* **COL** *cc* **LITERAL STRING IS TOO LONG. STRING BEGINS** *string*

Explanation: The character string constant that was referenced in this message is too long. The erroneous string begins with *string*. Most strings have these limits:

- For a hexadecimal constant (X, GX, or UX), the number of hexadecimal digits must not exceed 32704.
- Any other character string constant must be short enough so that its UTF-8 representation requires no more than 32704 bytes.
- Any other graphic string constant must be short enough so that its UTF-8 representation requires no more than 32704 bytes.

In some contexts, however, the limit is smaller; see the *DB2 SQL Reference* for the specific cases.

If this release of DB2 increases the maximum allowed length of a string constant, then an attempt to exploit the new length when the value of the NEWFUN option is NO (New Function Mode is NO) will produce this message.

For special registers, the allowable length depends on the particular special register. A string that exceeds the limit can be specified only through assignment from a host variable. If this release of DB2 increases the maximum allowed length of a string constant, then an attempt to exploit the new length before using the New Function Mode produces this SQL return code.

Two consecutive string delimiters are used to represent one string delimiter within the character string, but these count as 2 bytes when calculating the lengths of character string constants.

Severity: 8 (error)

System action: If the SQL statement containing the string is a declaration, it is discarded. If the SQL statement containing the string is executable, it is replaced by host language statements that, if executed, set an error code in the SQLCA.

User response: Correct the string. If a shorter string is acceptable, reduce the length of the string constant. If a long string is required, a host variable is required.

DSNH103I **E** *csectname* **LINE** *nnnn* **COL** *cc* **INVALID NUMERIC LITERAL** *token*

Explanation: The *token* was found. It begins with a digit, but it is not a valid integer, decimal, or float literal. Identifiers cannot begin with a digit except for the WHENEVER statement with HOST(FORTRAN), and the identifiers in COBOL.

Severity: 8 (error)

System action: If the containing SQL statement is a declaration, it is discarded. If the containing SQL statement is executable, it is replaced by host language statements that will, if executed, set an error code in the SQLCA.

User response: Check the indicated token for validity. Use apostrophes or quotation marks if a character literal is desired. Remove the invalid characters if a number is desired.

DSNH104I **E** *csectname* **LINE** *nnnn* **COL** *cc* **ILLEGAL SYMBOL** *"token"***. SOME SYMBOLS THAT MIGHT BE LEGAL ARE:** *token-list*

Explanation: A syntax error was detected where the symbol *"token"* occurs in the SQL statement. The list of symbols that might be legal shows some alternate symbols that could possibly be correct at that point, if the preceding part of the statement is entirely correct.

However, the preceding part of the statement might be incorrect. For example, if an important keyword is omitted, DB2 detects the error later, and not always immediately after the point where the keyword should appear. The list of alternate symbols are only suggestions.

Severity: 8 (error)

System action: If the containing SQL statement is a declaration, it is discarded. If the containing SQL statement is executable, it is replaced by host language statements that will, if executed, set an error code in the SQLCA.

User response: Correct the statement and precompile the program again.

DSNH105I **E** *csectname* **LINE** *nnnn* **COL** *cc*, **INVALID STRING CONSTANTS**

Explanation: Incorrect graphic or mixed data literals were found on line *nnnn*, column *cc*.

Severity: 8 (error)

System action: The literal is ignored and the SQL statement containing the problem is discarded.

User response: Specify the correct format of literals.

System programmer response: Check for a quote, an apostrophe, shift-out (so) or shift-in (si) characters, G or N characters, or the number of characters within the literal.

DSNH107I **E** *csectname* **LINE** *nnnn* **COL** *cc* **THE NAME** *name* **IS TOO LONG. MAXIMUM ALLOWABLE SIZE IS** *size*

Explanation: For each type of *name*, the *DB2 SQL Reference* lists the maximum allowed length.

If this release of DB2 increases the maximum allowed length of a name, then an attempt to exploit the new length when the value of the NEWFUN option is NO will produce this message.

Severity: 8 (error)

System action: If the SQL statement containing the problem is a declaration, it is discarded. If the SQL statement containing the problem is executable, it is replaced by host language statements that will, if executed, set an error code in the SQLCA.

User response: Correct the name length.

DSNH109I **E** *csectname* **LINE** *nnnn* **COL** *cc type* **CLAUSE NOT PERMITTED**

Explanation: The *type* is INTO, ORDER BY, FOR UPDATE, , USING, LARGE, or CORRELATION NAME.

- A SELECT INTO statement cannot include ORDER BY, because the result cannot be more than a single row.
- A SELECT INTO statement cannot include FOR UPDATE OF, because position is not retained.
- SELECT clauses used in cursor declarations or subselects cannot have an INTO clause because the INTO is on the FETCH statement.
- A SELECT INTO statement cannot specify KEEP UPDATE LOCKS, except in USE AND KEEP UPDATE LOCKS.
- CREATE VIEW statements cannot include any of the above clauses.
- The UNION operator is not allowed in an embedded SELECT.
- A CREATE TABLESPACE statement cannot specify LOCKPART without the NUMPARTS option.
- An ALTER TABLESPACE statement cannot specify LOCKPART for a non-partitioned table space.
- A table space with LOCKPART YES cannot be altered to LOCKSIZE TABLESPACE.
- A table space with LOCKSIZE TABLESPACE cannot be altered to LOCKPART YES.
- The LARGE clause without the NUMPARTS option is not allowed in a CREATE TABLESPACE statement.
- DESCRIBE INPUT statement can not have USING clause.
- The table being updated in a POSITIONED UPDATE statement can not be assigned a correlation name.

Severity: 8 (error)

System action: If the containing SQL statement is a declaration, it is discarded. If the containing SQL statement is executable, it is replaced by host language statements that will, if executed, set an error code in the SQLCA.

User response: Remove the clause. You might need to use a cursor to perform the processing. The ORDER and FOR UPDATE clauses are accepted on the DECLARE CURSOR statement. The INTO clause is accepted on the FETCH statement. The LARGE clause is accepted on a CREATE TABLESPACE statement with the NUMPARTS option.

DSNH110I **E** *csectname* **LINE** *nnnn* **COL** *cc* **INVALID HEX LITERAL BEGINNING** *string*

Explanation: The hexadecimal string constant that was referenced contains one or more of the following errors:

- one or more characters are not valid hexadecimal digits
- the number of digits is not even
- for a UX or GX string constant, the number of digits is not a multiple of 4

Severity: 8 (error)

System action: If the containing SQL statement is a declaration, it is discarded. If the containing SQL statement is executable, it is replaced by host language statements that will, if executed, set an error code in the SQLCA.

User response: Correct the string constant.

DSNH111W **W** *csectname* **LINE** *nnnn* **COL** *cc* **THE SUBPAGES OPTION IS NOT SUPPORTED FOR TYPE 2 INDEXES**

Explanation: You cannot use the SUBPAGES option for type 2 indexes.

Severity: 4 (warning)

System action: The option is ignored; processing continues.

Programmer response: Remove the SUBPAGES option to get rid of the warning.

DSNH113I **E** *csectname* **LINE** *nnnn* **COL** *cc* **INVALID CHARACTER FOUND IN** *string*, **REASON CODE -** *nnn*

Explanation: The string contains an invalid character. It can be an SQL ordinary identifier name, a host variable name, or a DBCS comment.

Certain SQL identifiers (for example names of buffer pools, databases, and plans) must contain only uppercase alphabetic or national characters and numerics; the first character must be alphabetic or national.

The following reason codes apply to SBCS identifiers:

000 An invalid character was found in the SBCS identifier (including the case in which a DBCS identifier was used where only an SBCS identifier is allowed).

The following reason codes apply to DBCS identifiers or comments:

101 An odd number of bytes exists between the shift-out and the shift-in character.

102 Either a shift-in or shift-out character is missing.

103 DBCS blanks X'4040' are not allowed.

104 There are no characters between the shift-out and the shift-in characters.

105 Shift-out cannot be the first byte of the DBCS character between the shift-out and the shift-in characters.

Severity: 8 (error)

System action: Processing is terminated.

User response: Correct the name.

DSNH115I **E** *csectname* **LINE** *nnnn* **COL** *cc* **ILLEGAL USE OF COMPARISON OPERATOR** *operator*

Explanation: The syntax is not correct for this type of comparison. A simple comparison such as '>' must not be followed by a list of items. The ANY and ALL comparisons must precede a subselect, rather than an expression or a list of items.

Severity: 8 (error)

System action: If the containing SQL statement is a declaration, it is discarded. If the containing SQL statement is executable, it is replaced by host language statements that will, if executed, set an error code in the SQLCA.

User response: Correct the syntax. If necessary, refer to Chapter 2 of *DB2 SQL Reference* for the correct syntax.

DSNH117I **E** *csectname* **LINE** *nnnn* **COL** *cc* **STATEMENT INSERTS THE WRONG NUMBER OF VALUES**

Explanation: The number of entries in a value-list is not the same as the number of columns specified in an INSERT statement.

Severity: 8 (error)

System action: Processing continues.

User response: Correct the syntax. If necessary, refer to Chapter 5 of *DB2 SQL Reference* for the correct syntax.

DSNH120I **E** *csectname* **LINE** *nnnn* **COL** *cc* **FUNCTION USAGE NOT PERMITTED:** *fname*

Explanation: A WHERE clause or SET clause contains an aggregate function, possibly as a result of a reference to a column that is derived from an aggregate function in a view definition. An aggregate function is allowed in a WHERE clause only if that clause appears within a subquery of a HAVING clause.

Severity: 8 (error)

System action: If the containing SQL statement is a declaration, it is discarded. If the containing SQL statement is executable, it is replaced by host language statements that will, if executed, set an error code in the SQLCA.

User response: Correct the syntax. A subselect clause might be needed to produce the function result. If necessary, refer to Chapter 4 of *DB2 SQL Reference* for the correct syntax.

DSNH125I **E** *csectname* **LINE** *nnnn* **COL** *cc* **INVALID ORDER BY COLUMN NUMBER:** *number*

Explanation: The identified column number is either less than 1 or greater than the number of items in the select-list.

Severity: 8 (error)

System action: The cursor declaration is discarded.

User response: Correct the column number. If necessary, refer to Chapter 4 of *DB2 SQL Reference* for the correct syntax.

DSNH126I **E** *csectname* **LINE** *nnnn* **COL** *cc* **ORDER BY CANNOT BE USED WITH A FOR UPDATE OF CLAUSE**

Explanation: The ORDER BY clause makes the cursor incapable of being updated.

Severity: 8 (error)

System action: The cursor declaration is discarded.

User response: This is a DB2 restriction. You can have either an ordered cursor or a cursor that can be updated. You cannot specify both attributes for the same cursor. You might need to change your application logic.

DSNH129I **E** *csectname* **LINE** *nnnn* **COL** *cc* **TOO MANY TABLES SPECIFIED IN SQL STATEMENT**

Explanation: In a subselect, (including all subqueries), there can be no more than 225 references to table names.

Severity: 8 (error)

System action: If the containing SQL statement is a declaration, it is discarded. If the containing SQL statement is executable, it is replaced by host language statements that will, if executed, set an error code in the SQLCA.

User response: Reduce the number of table references in each subselect. Each reference to a single table counts separately. The count only applies to table/view references in the FROM list. Each view is only counted one time. Refer to Chapter 4 of *DB2 SQL Reference* for the definition of a subselect.

DSNH130I **E** *csectname* **LINE** *nnnn* **COL** *cc* **THE ESCAPE CLAUSE '***x***' CONSISTS OF MORE THAN ONE CHARACTER**

Explanation: The literal *x* used in the ESCAPE clause is too long. Only one SBCS or one DBCS character is allowed.

Severity: 8 (error)

System action: The corresponding SQL statement is replaced by host language statements that will, if executed, set an error code in the SQLCA.

User response: Correct the string. The literal string for the ESCAPE clause can only be one SBCS or DBCS character.

DSNH131I **E** *csectname* **LINE** *nnnn* **STATEMENT WITH LIKE PREDICATE HAS INCOMPATIBLE DATA TYPES**

Explanation: If the column name at the left of LIKE or NOT LIKE is a character type, the expression at the right must be a character type. If the column name is a graphic type, the expression at the right must be a graphic type.

Severity: 8 (error)

System action: The statement is ignored.

User response: Check the data type of every operand.

DSNH132I **E** *csectname* **LINE** *nnnn* **COL** *cc* **INVALID LIKE PREDICATE**

Explanation: The value following LIKE must be an alphanumeric host variable or literal, graphic host variable or literal, or special character literal.

Severity: 8 (error)

System action: If the containing SQL statement is a declaration, it is discarded. If the containing SQL statement is executable, it is replaced by host language statements that will, if executed, set an error code in the SQLCA.

User response: Correct the syntax. If necessary, refer to Chapter 2 of *DB2 SQL Reference* for the correct syntax.

DSNH142I **E** *csectname* **LINE** *nnnn* **COL** *cc* **THE SQL STATEMENT IS NOT SUPPORTED**

Explanation: The *token* was found. An SQL statement
was detected that is not supported by the database. The
statement might be valid for other IBM relational
database products or it might be valid in another
context. For example, statements such as VALUES and
SIGNAL or RESIGNAL SQLSTATE can appear only in
certain contexts, such as inside a trigger body or in an
SQL Procedure.

Severity: 8 (error)

System action: The SQL statement is replaced by host language statements that will, if executed, set an error code in the SQLCA.

User response: Change the syntax of the SQL statement or remove the statement from the program.

DSNH170I **E** *csectname* **LINE** *nnnn* **COL** *cc* **NUMBER OF ARGUMENTS SPECIFIED FOR** *function-name* **IS INVALID**

Explanation: An SQL statement specified the scalar function *function-name* with either too many or too few arguments.

Severity: 8 (error)

System action: If the containing SQL statement is a declaration, it is discarded. If the containing SQL statement is executable, it is replaced by host language statements that will, if executed, set an error code in the SQLCA.

User response: Correct the function call. Refer to Chapter 3 of *DB2 SQL Reference* for information about the number of arguments required by the scalar function *function-name*.

DSNH171I **E** *csectname* **LINE** *nnnn* **COL** *cc* **THE DATA TYPE, LENGTH, OR VALUE OF ARGUMENT** *nn* **OF** *function-name* **IS INVALID**

Explanation: Either the data type, the length or the value of argument *nn* of scalar function *function-name* is incorrect.

Severity: 8 (error)

System action: If the containing SQL statement is a declaration, it is discarded. If the containing SQL statement is executable, it is replaced by host language statements that will, if executed, set an error code in the SQLCA.

User response: Correct the argument. Refer to Chapter 3 of *DB2 SQL Reference* for rules for each argument of the scalar function *function-name*.

DSNH182I **E** *csectname* **LINE** *nnnn* **COL** *cc* **AN ARITHMETIC EXPRESSION WITH A DATETIME VALUE IS INVALID**

Explanation: The specified arithmetic expression contains an improperly used date/time value or labeled duration.

Severity: 8 (error)

System action: If the containing SQL statement is a declaration, it is discarded. If the containing SQL statement is executable, it is replaced by host language statements that will, if executed, set an error code in the SQLCA.

User response: Correct the indicated expression. Refer to Chapter 2 of *DB2 SQL Reference* for the rules for date/time arithmetic.

DSNH184I **E** *csectname* **LINE** *nnnn* **COL** *cc* **AN ARITHMETIC EXPRESSION WITH A DATETIME VALUE CONTAINS A PARAMETER MARKER**

Explanation: The specified expression contains a parameter marker improperly used with a date/time value.

Severity: 8 (error)

System action: If the containing SQL statement is a declaration, it is discarded. If the containing SQL statement is executable, it is replaced by host language statements that will, if executed, set an error code in the SQLCA.

User response: Correct the indicated expression. Refer to Chapter 2 of *DB2 SQL Reference* for information about date/time arithmetic.

DSNH198I **E** *csectname* **LINE** *nnnn* **COL** *cc* **EMPTY SQL STATEMENT IGNORED**

Explanation: The referenced SQL statement has no text between the EXEC SQL and the statement terminator.

Severity: 8 (error)

System action: The statement is ignored.

User response: Correct the statement, or remove it.

DSNH199I **E** *csectname* **LINE** *nnnn* **COL** *cc* **INVALID KEYWORD** *keyword***; VALID SYMBOLS ARE:** *token list*

Explanation: A syntax error was detected at the keyword specified in the message. As an aid to the programmer, a partial list of valid tokens is provided in the message. Only those tokens that fit are listed. The error might be earlier in the statement, but the syntax of the statement appears to be valid up to this point.

Severity: 8 (error)

System action: If the containing SQL statement is a declaration, it is discarded. If the containing SQL statement is executable, it is replaced by host language statements that will, if executed, set an error code in the SQLCA.

User response: Check the SQL statement in the area of the indicated keyword. A colon or SQL delimiters might be missing. Correct the SQL statement.

DSNH203I **W** *csectname* **LINE** *nnnn* **COL** *cc* **STATEMENT REFERENCES COLUMN** *column name***, WHICH IS DECLARED IN MORE THAN ONE OF THE SPECIFIED TABLES**

Explanation: The column *column name* was referenced without sufficient qualification. Type checking cannot be done.

Severity: 4 (warning)

System action: Processing continues.

User response: If the table declares are correct and the column name is not unique, this creates an error in the BIND step. Correct the DECLARE TABLE statement, or use the correct table name to qualify the column name.

DSNH204I **W** *csectname* **LINE** *nnnn* **COL** *cc* **STATEMENT REFERENCES UNDECLARED TABLE** *table-name*

Explanation: The table definition was not provided. It is optional, but checking cannot be done without the table information. (BIND checks against the actual table during later processing.)

Severity: 4 (warning)

System action: Processing continues.

DSNH205I **W** *csectname* **LINE** *nnnn* **COL** *cc* **COLUMN** *column-name* **WAS NOT DECLARED IN TABLE** *table-name*

Explanation: The column name *column-name* is not contained in the table declarations. The name specified in a DISTINCT function must reference a column of the table. The value expression of an ALL function must reference a column of the table.

Severity: 4 (warning)

System action: Processing continues.

User response: If the table declares are correct and the column name is not in the table, this creates an error in the BIND step. Correct the DECLARE TABLE statement or the column name.

DSNH206I **W** *csectname* **LINE** *nnnn* **COL** *cc* **STATEMENT REFERENCES COLUMN** *column name* **WHICH IS NOT DECLARED IN THE SPECIFIED TABLE(S)**

Explanation: The column *column name* is not contained in any TABLE declaration for the tables specified in the statement.

Severity: 4 (warning)

System action: Processing continues.

User response: If the table declares are correct and the column name is not in the tables, this creates an error in the BIND step. Correct the DECLARE TABLE statements or the column name.

System programmer response: Check the spelling of the column name. Ensure that all the tables needed were specified in the statement.

DSNH207I **E** *csectname* **LINE** *nnnn* **COL** *cc* **ILLEGAL ORDER BY SPECIFICATION**

Explanation: If the select expression involves a UNION operator, the ORDER BY list must use numeric constants rather than variables.

Severity: 8 (error)

System action: If the containing SQL statement is a declaration, it is discarded. If the containing SQL statement is executable, it is replaced by host language statements that will, if executed, set an error code in the SQLCA.

User response: Correct the syntax. If necessary, refer to Chapter 4 of *DB2 SQL Reference* for the correct syntax.

DSNH251I **E** *csectname* **LINE** *nnnn* **COL** *cc* **TOKEN** *name* **IS NOT VALID**

Explanation: The location name cannot contain alphabetic extenders for national languages (that is, #, @, and $ in the United States).

Severity: 8 (error)

System action: Processing continues.

User response: Correct the content of the location name.

DSNH290I **E** *csectname* **LINE** *nnnn* **COL** *cc* **NO VALID 'END DECLARE SECTION' FOUND**

Explanation: The user's source program contains a BEGIN DECLARE SECTION, but no corresponding END DECLARE SECTION was found.

Severity: 8 (error)

System action: Processing continues.

User response: Check the application program and ensure that a valid END DECLARE SECTION statement is present.

DSNH291I **E** *csectname* **LINE** *nnnn* **COL** *cc* **NO VALID 'BEGIN DECLARE SECTION' FOUND**

Explanation: The user's source program contains an END DECLARE SECTION, but no corresponding BEGIN DECLARE SECTION was found.

Severity: 8 (error)

System action: Processing continues.

User response: Check the application program and ensure that a valid BEGIN DECLARE SECTION is present.

DSNH292I **E** *csectname* **LINE** *nnnn* **COL** *cc* **HOST NAME** *name* **IS TOO LONG; MAXIMUM IS** *size* **CHARACTERS**

Explanation: Fully qualified C host variable names cannot contain more than 256 characters.

Severity: 8 (error)

System action: If the containing SQL statement is executable, it is replaced by host language statements that will, if executed, set an error code in the SQLCA.

User response: Reduce the length of names or the level of qualification of the host variable name in error.

DSNH303I **W** *csectname* **LINE** *nnnn* **COL** *cc* **STATEMENT SELECTS INCOMPATIBLE DATA TYPE INTO** *hostvar*

Explanation: The data type of host variable *hostvar* is not compatible with the apparent type of the corresponding select expression. One of them is numeric and the other is character, or one of them is character and the other is graphic. If the select expression is a column reference, the data type is obtained from its declaration.

Severity: 4 (warning)

System action: Processing continues.

User response: Ensure that the table definitions are current and that the host variable has the proper data type.

System programmer response: Ensure that the table definitions are current, and that the host variable has the proper data type.

DSNH307I **E** *csectname* **HOST VARIABLE ALREADY DECLARED**

Explanation: The host variable being registered with SQLGALHV has already been registered.

Severity: 8 (error)

System action: The host variable is ignored.

User response: Continue the compilation or precompilation process.

DSNH308I **U** *csectname* **MAXIMUM NUMBER OF HOST VARIABLES EXCEEDED**

Explanation: The registration of host variables exceeded the internal symbol table storage capacity.

Severity: 16 (unrecoverable error)

System action: Further processing halts.

User response: Terminate compilation or precompilation.

DSNH310I **W** *csectname* **LINE** *nnnn* **COL** *cc language* **HOST VARIABLE** *name* **WAS DECLARED IN** *location*

Explanation: Host variable *name* should not be declared in this *location* of the source program. This message is issued only at reference time.

Severity: 4 (warning)

System action: Processing continues.

User response: If this host variable is used, the user assumes responsibility for possible exposure of incorrect output. Refer to Part 2 of *DB2 Application Programming and SQL Guide* to determine where host variables can be validly declared.

Declaring a COBOL host variable in the File Section might lead to incorrect output from SQL statements or abend 04E with RC=00E7000C.

DSNH312I **E** *csectname* **LINE** *nnnn* **COL** *cc* **UNDEFINED OR UNUSABLE HOST VARIABLE** *name*

Explanation: The *name* in the message is used in an SQL statement, but either no declaration was found in the application program so far, or the attributes are improper for this use. Declarations for host variables used in a DECLARE CURSOR statement must precede the SQL statement, regardless of the ONEPASS or TWOPASS option for the precompiler, or a parameter marker was expected in an EXECUTE IMMEDIATE statement. This message is issued only at reference time.

Severity: 8 (error)

System action: If the containing SQL statement is a declaration, it is discarded. If the containing SQL statement is executable, it is replaced by host language statements that will, if executed, set an error code in the SQLCA.

User response: Check the declaration and the attributes of the named host variable. If you intended the name to refer to a column, the colon (:) should not be present. If you intended a literal, you need to enclose the value in string delimiters (apostrophes or quotation marks).

If you intended to use a host variable, only a subset of the possible variable declarations are valid for use as a DB2 host variable. The rules for host variables are listed in the application programming guides. Use the cross-reference listing and precompiler source listing to determine the source of the problem.

- If you use the STDSQL(YES) option for the precompilation, make sure that all of the host variable declarations are declared within BEGIN/END DECLARE SECTIONs.
- If the cross-reference listing does not contain the host variable name, the declaration is missing. If the source for a declaration is present, a missing terminator from a comment or from a string might cause this type of error. The margins could also cause this type of error.
- If the name is present, but the data type is not listed, the host variable is not usable; it does not follow the rules outlined for valid host variable definitions.
- If several definitions exist for the name, the first reference might occur before the definition. The use of the TWOPASS option helps by processing DECLARE statements before the other types of statements. However, host variable declarations and cursor declarations are processed in the same pass. The host variables referenced in cursor declarations must be defined before the cursor is defined in the application program.
- A parameter marker is expected rather than a host variable in an EXECUTE IMMEDIATE statement.
- The data type of a host variable passed in a CALL statement is not character or varying character.
- The length attribute of a character or varying character host variable passed in a CALL statement exceeds 255 characters.

DSNH313I **E** *csectname* **LINE** *nnnn* **COL** *cc* **INVALID TYPE OR LENGTH OF HOST VARIABLE '***name***'**

Explanation: The host variable *name* is used in an SQL statement, but either the type or the length is improper for this use.

Severity: 8 (error)

System action: The corresponding SQL statement is replaced by host language statements that will, if executed, set an error code in the SQLCA.

User response: Check the declaration and the attributes of the named host variable. Refer to Part 2 of *DB2 Application Programming and SQL Guide* for the allowable host variable type and length.

DSNH314I **E** *csectname* **LINE** *nnnn* **COL** *cc* **REFERENCE TO HOST VARIABLE** *name* **IS AMBIGUOUS**

Explanation: The host variable *name* was defined more than once in this application program. The precompiler cannot distinguish which host variable definition should be used here. The following table indicates some possible combinations. DB2 supports only a single adjacent qualifier for a host variable name.

Case	Ref	Definition	Result	Suggested Change
1	a	x.a	Valid	None
2	x.a	x.a, y.a	Valid	None
3	a	x.a, y.a	DSNH314I	Reference x.a or y.a
4	a	x.a, a	Valid	None
5	x.a	q.x.a, v.x.a	No precompiler error, but host language compiler will issue an error.	Variable name changes
6	a	a, a	No precompiler error, but host language compiler will issue an error.	Variable name changes

Severity: 8 (error)

System action: If the SQL statement is a declaration, it is discarded. If the containing SQL statement is executable, it is replaced by host language statements that will, if executed, set an error code in the SQLCA.

User response: Make the host variable name unique or use qualification to indicate which host variable definition is to be used, as suggested in the above table.

DSNH330I **E** *csectname* **LINE** *nnnn* **COL** *cc* **TEXT COULD NOT BE CONVERTED TO CCSID 1208, REASON** *reason-code*, **CHARACTER** *code-point*

Explanation: DB2 parses SQL statements in Unicode UTF-8, which is also known as CCSID 1208. DB2 will convert the application program internally (not externally) to UTF-8 for processing if an application program uses a different CCSID (for example, an EBCDIC CCSID).

Text from the application program could not be converted from the source CCSID to CCSID 1208. The reason code indicates the context of this error:

012 Invalid code point. An invalid character was sent to be converted as indicated by the code-point

016 Other invalid input character. For example, this could be a mismatch between shift-in and shift-out, or the absence of half of a DBCS character. The code-point is blank.

Severity: 8 (error)

System action: The statement is discarded.

Programmer response: If the reason code is 012, change the text in the application program to a valid code point. If the reason code is 016, change the text in the application program to conform to the rules for well-formed mixed data.

DSNH331I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **SUBSTITUTION OCCURRED ON CONVERSION FROM CCSID** *from-ccsid* **TO** *to-ccsid*

Explanation: During conversion of the source text into UTF-8 (CCSID1208), a character was encountered that resulted in a substitution character of either '3F'x (in EBCDIC code pages), or '1A'x (in UTF-8).

Severity: 4 (warning)

System action: Processing of the statement continues.

User response: Review the character that was translated to determine if the substitution character '3F'x is correct. If it is correct, then the message can be ignored. Otherwise, change the character to the correct code point.

DSNH332I **S** *csectname* **CHARACTER CONVERSION FROM CCSID** *from-ccsid* **TO CCSID 1208 IS NOT SUPPORTED, REASON** *reason-code*

Explanation: DB2 parses SQL statements in Unicode UTF-8, which is also known as CCSID 1208. DB2 will convert the application program internally (not externally) to UTF-8 for processing if an application program uses a different CCSID (for example, an EBCDIC CCSID). The source CCSID can be specified by the CCSID option.

This message indicates that conversion could not take place from the source CCSID to CCSID 1208. The reason code indicates the context of this error:

000 The value of the CCSID option is neither 65535, nor in the range 1 through 65533.

001 No conversion is defined from the source CCSID to CCSID 1208.

002 No conversion is defined from CCSID 1208 to the source CCSID.

Severity: 12 (severe error)

System action: The precompiler is terminated.

User response: Check that the value of the CCSID option is valid. Run the precompilation again with a valid CCSID if you do not want to use the EBCDIC system CCSID.

System programmer response: Make sure that the Unicode System Services conversion image includes the required conversions. See the V2R8.0-V2R10.0 *z/OS Support for Unicode: Using Conversion Services*.

DSNH350I **E** *csectname* **LINE** *nnnn* **COL** *cc*. **INVALID SPECIFICATION OF A LARGE OBJECT COLUMN**

Explanation: The ALTER TABLE, CREATE TABLE, or CREATE INDEX statement is invalid for one of the following reasons:

- A LOB column cannot be added to a temporary table.
- A LOB column cannot be added to a table defined with an EDITPROC.
- The *PRIMARY KEY* clause cannot specify a LOB column as a column of the primary key.
- The *UNIQUE* clause cannot specify a LOB column as a column of the unique key.
- The *referential-constraint* clause cannot specify a LOB column as a column of a foreign key.
- The CREATE INDEX statement cannot name a LOB column as a column of the index key.
- A LOB column cannot be specified in a *references* clause.
- A *table check constraint* cannot be specified in the definition of a LOB column.

Severity: 8 (error)

System action: The statement cannot be executed.

User response: Correct and reissue the failing statement.

DSNH372I **E** *csectname* **LINE** *nnnn* **COL** *cc*. **ONLY ONE COLUMN DEFINED AS ROWID IS ALLOWED IN A TABLE**

Explanation: An attempt was made to create a table with more than one ROWID column or to add a ROWID column to a table which already has one.

Severity: 8 (error)

System action: The statement cannot be executed.

User response: For a CREATE TABLE statement, select only one column to have the row ID data type. For an ALTER TABLE statement, a ROWID column already exists for the table. Do not attempt to add another column with data type row ID to the table.

DSNH385I **W** *csectname* **LINE** *nnnn* **COL** *cc* **ASSIGNMENT TO AN SQLSTATE OR SQLCODE VARIABLE IN AN SQL PROCEDURE MAY BE OVER-WRITTEN AND DOES NOT ACTIVATE ANY HANDLER**

Explanation: The SQL procedure includes at least one statement that assigns a value to the SQLSTATE or SQLCODE special variables. Since DB2 assigns values to those variables to indicate the results of SQL statement execution, DB2 overwrites the values assigned by the SQL assignment statement.

An SQL procedure cannot activate a handler by assigning a value to SQLSTATE.

Severity: 4 (warning)

System action: Processing continues.

User response: To prevent this warning, do not assign values to the SQLCODE or SQLSTATE special variables.

DSNH401I **W** *csectname* **LINE** *nnnn* **COL** *cc* **STATEMENT SPECIFIES INCOMPATIBLE OPERANDS FOR COMPARISON OPERATOR** *name*

Explanation: The comparison operation *name* is used with a mixture of incompatible data type operands.

Severity: 4 (warning)

System action: Processing continues.

User response: Make both operands compatible character data types. Ensure that the table declaration is correct.

DSNH402I **W** *csectname* **LINE** *nnnn* **COL** *cc* **STATEMENT APPLIES** *name* **TO CHARACTER DATA**

Explanation: The numeric operation or function *name* is used with a non-numeric operand.

Severity: 4 (warning)

System action: Processing continues.

User response: Make the operand numeric or use a character function. Ensure that the table declaration is correct.

DSNH404I **W** *csectname* **LINE** *nnnn* **COL** *cc* **STATEMENT INSERTS OR UPDATES COLUMN** *name* **WITH VALUE THAT IS TOO LONG**

Explanation: An INSERT or UPDATE statement specifies a value that is longer than the maximum length string that can be stored in that column, according to the DECLARE TABLE statement.

Severity: 4 (warning)

System action: Processing continues.

User response: Ensure that the length is allowed and that the table declaration is correct.

DSNH405I **E** *csectname* **LINE** *nnnn* **COL** *cc* **LITERAL** *number* **IS OUT OF RANGE**

Explanation: The number is not in the proper range:
- +5.4E−79 to 7.2E+75 for floating-point
- $-(10^{31}-1)$ to $+(10^{31}-1)$ for packed decimal
- −2147483648 to 2147483647 for integers

Severity: 8 (error)

System action: If the containing SQL statement is a declaration, it is discarded. If the containing SQL statement is executable, it is replaced by host language statements that will, if executed, set an error code in the SQLCA.

User response: Correct the number.

DSNH408I **W** *csectname* **LINE** *nnnn* **COL** *cc* **STATEMENT INSERTS OR UPDATES COLUMN** *name* **WITH INCOMPATIBLE DATA TYPE**

Explanation: The data type of the value to be inserted into column *name* by an INSERT or UPDATE statement is incompatible with the declared data type of the column. Both must be numeric or both must be character.

Severity: 4 (warning)

System action: Processing continues.

Programmer response: Be sure that the table definitions are current, and that host variables and literals have the proper data types and are matched.

DSNH414I **E** *csectname* **LINE** *nnnn* **COL** *cc* **THE NUMERIC OR DATETIME COLUMN** *column-name* **IS SPECIFIED IN A LIKE PREDICATE**

Explanation: Column *column-name* with a numeric data type was specified in a LIKE predicate. Columns specified in a LIKE predicate must be character or graphic string columns.

Severity: 8 (error)

System action: If the containing SQL statement is a declaration, it is discarded. If the containing SQL statement is executable, it is replaced by host language statements that will, if executed, set an error code in the SQLCA.

User response: Correct the statement.

DSNH434I **OPTION** *keyword* **IS A DEPRECATED FEATURE**

Explanation: *keyword* is a deprecated feature that will not be supported in releases following DB2 Version 8. It is accepted, but we recommend that you discontinue the use of this keyword.

For indexes, we recommend that you use type 2 indexes rather than type 1 indexes.

Severity: 4 (warning)

System action: Processing continues normally.

User response: No change is required for the current release. However, we recommend that you change your SQL statement to remove this feature to prepare for future releases when this feature is not supported.

DSNH435I **E** *csectname* **LINE** *nnnn* **COL** *cc* **AN INVALID SQLSTATE** *sqlstate* **IS SPECIFIED IN THE FUNCTION RAISE_ERROR OR IN A SIGNAL OR RESIGNAL SQLSTATE STATEMENT**

Explanation: The SQLSTATE specified in the RAISE_ERROR function or specified in a SIGNAL or RESIGNAL SQLSTATE statement does not conform to the rules for an application defined SQLSTATE.

Severity: 8 (error)

System action: The statement cannot be processed.

User response: Correct the SQLSTATE specified in the RAISE_ERROR function or SIGNAL or RESIGNAL statement. The SQLSTATE must be a character string containing exactly 5 characters. It must be of type CHAR defined with a length of 5, or a type VARCHAR defined with a length of 5 or greater. The SQLSTATE value must follow the rules for application-defined SQLSTATEs as follows:
- Each character must be from the set of digits ('0' through '9') or non-accented upper case letters ('A' through 'Z').
- The SQLSTATE class (first two characters) cannot be '00' since this represents successful completion.

DSNH449I **W** *csectname* **LINE** *nnnn* **COL** *cc* **CREATE FUNCTION OR PROCEDURE FOR** *routine-name* **CONTAINS AN INVALID FORMAT OF THE EXTERNAL NAME CLAUSE OR IS MISSING THE EXTERNAL NAME CLAUSE**

Explanation: An error was found in the EXTERNAL
NAME clause of the CREATE FUNCTION or CREATE
PROCEDURE statement for *name*, or the clause is
needed but was not specified. See SQLCODE -449 for
specific information about rules for LANGUAGE JAVA.

Severity: 8 (error)

System action: The corresponding SQL statement is replaced by host language statements that will, if executed, set an error code in the SQLCA.

User response: Correct the syntax of the SQL statement. Refer to the *DB2 SQL Reference* for information on the EXTERNAL NAME clause.

DSNH457I **E** *csectname* **LINE** *nnnn* **COL** *cc*. **A FUNCTION OR DISTINCT TYPE CANNOT BE CALLED** *name* **SINCE IT IS RESERVED FOR SYSTEM USE.**

Explanation: The user-defined function or distinct type cannot be created or referenced because the name selected is reserved for use by the system.

A number of names used as keywords are reserved for system use. These names may not be used as user-defined functions or distinct-type-names, **even if they are delimited identifiers**. These names are:

```
=        <        >        >=       <=
<>       ¬=       ¬>       ¬>
ALL      AND      ANY      BETWEEN  DISTINCT
EXCEPT   EXISTS   FALSE    FOR      FROM
IN       IS       LIKE     MATCH    NOT
NULL     ONLY     OR       OVERLAPS SIMILAR
SOME     TABLE    TRUE     TYPE     UNIQUE
UNKNOWN
```

Severity: 8 (error)

System action: The statement cannot be executed.

User response: Select a name for the function or distinct type that is not reserved for system use.

DSNH473I **E** *csectname* **LINE** *nnnn* **COL** *cc*. **A USER DEFINED DATA TYPE CANNOT BE CALLED THE SAME NAME AS A SYSTEM PREDEFINED TYPE (BUILT-IN TYPE).**

Explanation: The name of a data type to be created has an unqualified name that is the same as a system-predefined data type or is BOOLEAN. This is not allowed. Adding delimiters does not make the name valid. The following names are restricted:

```
BIGINT     BINARY    BLOB      BOOLEAN   CHAR
CHARACTER  CLOB      DATALINK  DATE      DBCLOB
DEC        DECIMAL   DOUBLE    FLOAT     GRAPHIC
INT        INTEGER   NUMERIC   REAL      REF
ROWID      SMALLINT  TIME      TIMESTAMP VARCHAR
VARGRAPHIC
```

Severity: 8 (error)

System action: The statement cannot be executed.

User response: Correct the statement to use another identifier for the name of the new user-defined type.

DSNH487I **E** *csectname* **LINE** *nnnn* **COL** *cc object-type object-name* **ATTEMPTED TO EXECUTE AN SQL STATEMENT WHEN THE DEFINITION OF THE FUNCTION OR PROCEDURE DID NOT SPECIFY THIS ACTION**

Explanation: SQL procedure *object-name* contains SQL statements, but the SQL procedure definition specifies the NO SQL option.

System action: The SQL statement cannot be executed.

Programmer response: Change the definition of the SQL procedure to allow SQL statements, or remove the failing SQL statement from the SQL procedure.

DSNH490I **E** *csectname* **LINE: MV.NNNN COL** *cc* **NUMBER** *number* **DIRECTLY SPECIFIED IN A SQL STATEMENT IS OUTSIDE THE RANGE OF ALLOWABLE VALUES IN THIS CONTEXT (***minval***,** *maxval***)**

Explanation: A number (*number*) was specified that is not valid in the context in which it was specified. The minimum allowed value in this context is *minval*. The maximum allowed value in this context is *maxval*. *n* must be within the range specified by *minval* and *maxval* (*minval* =< *n* =< *maxval*).

Severity: 8 (error)

System action: The statement was not executed.

User response: Change the value *n* to a valid value in the statement.

DSNH491I **E** *csectname* **LINE** *nnnn* **COL** *cc*. **CREATE STATEMENT FOR USER-DEFINED FUNCTION** *function-name* **MUST HAVE A RETURNS CLAUSE, AND EITHER THE EXTERNAL CLAUSE (WITH OTHER REQUIRED KEYWORDS) OR THE SOURCE CLAUSE.**

Explanation: A required clause is missing in the CREATE for function *function-name*. If EXTERNAL was specified, the following must also be specified:
- LANGUAGE
- PARAMETER STYLE

Severity: 8 (error)

System action: The statement cannot be executed.

User response: Correct and reissue the failing statement.

DSNH504I **E** *csectname* **LINE** *nnnn* **COL** *cc* **CURSOR** *name* **WAS NOT DECLARED**

Explanation: Identifier *name* is used as a cursor, but was not declared. The declaration might have been discarded because of errors.

Severity: 8 (error)

System action: If the containing SQL statement is a declaration, it is discarded. If the containing SQL statement is executable, it is replaced by host language statements that will, if executed, set an error code in the SQLCA.

User response: Correct or add the required cursor declaration.

DSNH506I **E** *csectname* **LINE** *nnnn* **COL** *cc* **DECLARE CURSOR FOR STATEMENT NAME MUST PRECEDE THE PREPARE STATEMENT FOR THAT STATEMENT NAME**

Explanation: This DECLARE *name* CURSOR statement is preceded by a PREPARE *name* statement. In DB2, declarations must precede action statements. This is not the same as in SQL/DS.

Severity: 8 (error)

System action: The declaration is noted as being in error and discarded.

User response: Move the DECLARE or PREPARE statement so that they are in the proper order, or specify the TWOPASS option if the host language is PLI, ASM, or C. The TWOPASS option has the precompiler check for declarations on the first pass and avoids this problem.

DSNH520I **E** *csectname* **THE OPEN STATEMENT FOR CURSOR cursor-name IS INVALID BECAUSE THE CURSOR WAS DEFINED BY AN ALLOCATE CURSOR STATEMENT**

Explanation: The cursor was not successfully declared, so it was assumed to be defined by an ALLOCATE CURSOR statement. A cursor defined by an ALLOCATE CURSOR statement is open after successful completion of the statement. Attempting to OPEN an allocated cursor is an error.

Severity: 8 (error)

System action: The statement cannot be executed.

User response: Add or fix the declaration of the cursor. If the cursor is supposed to be defined by an ALLOCATE CURSOR statement then remove the OPEN statement.

DSNH521I **W** *csectname* **THE statement FOR CURSOR cursor-name IS ASSUMED TO APPLY TO A DYNAMIC ALLOCATE CURSOR STATEMENT.**

Explanation: A FETCH, CLOSE, or DESCRIBE for an undeclared cursor was issued. The cursor is assumed to be defined using a dynamic ALLOCATE CURSOR statement.

Severity: 4 (warning).

System action: The cursor is assumed to be allocated dynamically.

User response: Ensure that the cursor in the FETCH, CLOSE, or DESCRIBE statement is dynamically allocated.

DSNH522I **E** *csectname* **DESCRIBE CURSOR cursor-name IS INVALID BECAUSE THE CURSOR WAS DEFINED BY A DECLARE CURSOR STATEMENT.**

Explanation: DESCRIBE CURSOR is allowed for cursors defined using the ALLOCATE CURSOR statement.

Severity: 8 (error).

System action: The statement cannot be executed.

User response: Remove the DESCRIBE CURSOR statement. Issue a DESCRIBE of the statement used to define the declared cursor.

DSNH526I **ERROR: DSNHDECP SPECIFIES AN UNDEFINED (0) VALUE FOR** *ccsid-parm***. NOTIFY THE SYSTEM PROGRAMMER.**

Explanation: The DSNHDECP module used by the DB2 precompiler contains an undefined (0) value for the CCSID parameter indicated by *ccsid-parm*. Use of undefined CCSIDs are not allowed by DB2.

System action: Processing terminates.

User response: Notify the system programmer.

System programmer response: The DSNHDECP needs to specify a valid, nonzero default CCSID (Coded Character Set Identifier) for each of the three supported encoding schemes: ASCII, EBCDIC, and UNICODE. Processing character data with an undefined CCSID can lead to unpredictable results, including loss or corruption of the data. A default CCSID must be defined even for encoding schemes not used on your DB2 system.

If *ccsid-parm* is ASCCSID, the default CCSID for single-byte ASCII-encoded character data is undefined (set to 0). If *ccsid-parm* is SCCSID, the default CCSID for single-byte EBCDIC-encoded character data is undefined (set to 0).

You need to generate a DSNHDECP module with appropriate CCSIDs for eachencoding scheme. See the *DB2 Installation Guide* for more information about CCSIDs.

DSNH527I WARNING: THE PRECOMPILER ATTEMPTED TO USE THE DB2-SUPPLIED DSNHDECP MODULE.

Explanation: The DSNHDECP module that is supplied by DB2 is being used to run the DB2 precompiler. The precompiler requires site-dependent settings that are specified in the DSNHDECP module that was created when DB2 was installed. The installation-generated DSNHDECP module typically resides in the SDSNEXIT library.

System action: Processing halts.

User response: Check the STEPLIB, JOBLIB, or system linbklist concatenation used in the DB2 address spaces. To ensure that the proper DSNHDECP module is used, verify that the library that contains the DSNHDECP module that is generated during installation precedes the SDSNLOAD library that contains the DSNHDECP module that was supplied by DB2.

System programmer response: The following actions are required to enable the DSNH CLIST to load your user-supplied DSNHDECP module rather than the DB2 DSNHDECP module that was supplied by DB2:

1. The JOBLIB or STEPLIB concatenation of any job or TSO userid that calls DSNH needs to allocate the library where the user-supplied DSNHDECP module resides (usually prefix.SDSNEXIT) ahead of the library where the DB2-supplied DSNHDECP module resides (prefix.SDSNLOAD).
2. The DSNH call should include the PCLOAD parameter, and it should be coded as shown here: PCLOAD('*(DSNHPC)').

DSNH557I E *csectname* LINE *nnnn* COL *cc* INCONSISTENT GRANT/REVOKE KEYWORD *keyword*; PERMITTED KEYWORDS ARE *keyword list*

Explanation: The identified *keyword* is not permitted. The *keyword list* is a list of the keywords that are permitted in this context. The combination of privileges requested on a GRANT or REVOKE statement is not allowed. The authorities must all be of one type and consistent with the form of GRANT or REVOKE. REVOKE UPDATE(column list) is not allowed; only REVOKE UPDATE is permitted.

Severity: 8 (error)

System action: The containing SQL statement is replaced by host language statements that will, if executed, set an error code in the SQLCA.

User response: Correct the syntax. If necessary, refer to Chapter 5 of *DB2 SQL Reference* for the correct syntax.

DSNH558I E *csectname* LINE *nnnn* COL *ccc* INVALID CLAUSE OR COMBINATION OF CLAUSES ON A GRANT OR REVOKE

Explanation: "PUBLIC AT ALL LOCATIONS" is invalid for non-table privilege GRANT or REVOKE statements. It is also invalid for table privilege GRANT or REVOKE statements that specify the ALTER or INDEX privilege.

Severity: 8 (error)

System action: The statement cannot be executed.

User response: Refer to Chapter 5 of *DB2 SQL Reference* for valid keywords for the GRANT or REVOKE statement.

DSNH577I E *csectname* LINE *nnnn* COL *cc object-type object-name* ATTEMPTED TO MODIFY DATA WHEN THE DEFINITION OF THE FUNCTION OR PROCEDURE DID NOT SPECIFY THIS ACTION

Explanation: An SQL procedure contains SQL statements that modify data, but the SQL procedure is defined with the NO SQL, READS SQL DATA, or CONTAINS SQL option.

Severity: 8 (error)

System action: The SQL statement cannot be executed.

User response: Change the definition of the SQL procedure to allow SQL statements that modify data, or remove the failing SQL statement from the SQL procedure.

DSNH580I E *csectname* LINE *nnnn* COL *cc* THE RESULT-EXPRESSIONS OF A CASE EXPRESSION CANNOT ALL BE NULL

Explanation: There is a CASE expression in the statement that has all the *result-expressions* (expressions following the THEN and ELSE keywords) coded with the keyword NULL.

Severity: 8 (error)

System action: The statement was not processed.

Programmer response: Change the CASE expression to include at least one *result-expression* with a keyword other than NULL.

DSNH581I E *csectname* LINE *nnnn* COL *cc* THE DATA TYPES OF THE RESULT-EXPRESSIONS OF A CASE EXPRESSION ARE NOT COMPATIBLE

Explanation: A CASE expression in the statement has

at least one *result-expression* (expressions following the THEN and ELSE keywords) whose data type does not match that of one of the other result-expressions.

Severity: 8 (error)

System action: The statement was not processed.

Programmer response: Change the CASE expression so that the data types of all of the *result-expressions* are compatible.

DSNH582I **E** *csectname* **LINE** *nnnn* **COL** *cc* **THE SEARCH-CONDITION IN A SEARCHED-WHEN-CLAUSE CANNOT BE A QUANTIFIED PREDICATE, IN PREDICATE, OR AN EXISTS PREDICATE**

Explanation: The *search-condition* in a *searched-when-clause* specifies a quantified predicate, an IN predicate or an EXISTS predicate, but is not allowed.

Severity: 8 (error)

System action: The statement was not processed.

Programmer response: Correct the *search-condition*.

DSNH585I **W** *csectname* **LINE** *nnnn* **COL** *cc* **DUPLICATE** *schema name* **SCHEMA NAME**

Explanation: The identified schema name was specified twice in the statement.

Severity: 4 (error)

System action: The duplicated schema name is ignore.

User response: Remove/correct the duplicate specification.

DSNH586I **E** *csectname* **LINE** *nnnn* **COL** *cc*. **THE TOTAL LENGTH OF THE CURRENT PATH SPECIAL REGISTER CANNOT EXCEED 254 CHARACTERS.**

Explanation: The CURRENT PATH special register is defined as a VARCHAR(254). The content of the string includes each schema name delimited with double quotes and separated from the next schema name by a comma. The total length of the string of all schema names in the CURRENT PATH cannot exceed 254 characters. A SET CURRENT PATH statement causing this message would exceed this limit.

Severity: 8 (error)

System action: The statement cannot be executed.

User response: Remove schema names to reduce the total length to fit the 254 character maximum length. If all the schema names are required, it may be necessary to consolidate some user-defined functions so that fewer schema names are required for the CURRENT PATH.

DSNH587I **A list of item-references are not in the same family.**

Explanation: Each item-reference in the SET assignment statement is either a host-variable or a transition-variable. The list of item-references must be of the same family, meaning if one of the item-references is a transition-variable, then all of the item-references in the list must be a transition-variable. If the statement is used in the triggered action of a CREATE TRIGGER statement, each item-reference must identify a transition variable. If the statement is used in any other context, each item-reference must identify a host variable.

Severity: 8 (error)

System action: The corresponding SQL statement is replaced by host language statements that will, if executed, set an error code in the SQLCA.

Programmer response: Correct the statement and execute it again.

User response: Correct the syntax of the SQL statement and precompile the program again. Refer to the DB2 SQL Reference to information on the SET assignment statement.

DSNH590I **E** *csectname* **LINE** *nnnn* **COL** *cc* **NAME** *name* **IS NOT UNIQUE IN ROUTINE** *routine-name*

Explanation: Name *name* is a parameter, SQL variable, condition name, or label in SQL procedure *routine-name*. *name* is not unique.

System action: The statement cannot be processed.

Severity: 8 (error)

User response: Change the name to make it unique within the routine.

DSNH599I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **COMPARISON FUNCTIONS ARE NOT CREATED FOR A DISTINCT TYPE BASED ON A LONG STRING DATA TYPE.**

Explanation: Comparison functions are not created for a distinct type based on a long string data type (BLOB, CLOB, DBCLOB, LONG VARCHAR, LONG VARGRAPHIC) since the corresponding function are not available for these built-in data types.

Severity: 4 (warning)

System action: Processing continues.

User response: No action is required.

DSNH612I **E** *csectname* **LINE** *nnnn* **COL** *cc* *column-name* **IS A DUPLICATE COLUMN NAME**

Explanation: The CREATE INDEX, CREATE TABLE, CREATE VIEW or ALTER TABLE statement specifies the same 'column-name' for two (or more) columns of the index, table, view, or the UPDATE OF clause of a trigger definition specifies the same column name more than once. Column names must be unique within an index, a table, a view, or in the UPDATE OF clause of a trigger definition. A column cannot be specified in more than one ALTER TABLE clause except if it is specified in an ALTER COLUMN clause and ADD CHECK CONSTRAINT clause.

Severity: 8 (error)

System action: The statement cannot be executed.

User response: Correct the CREATE statement to specify unique names for each of the columns of the index, table, view, or the columns in the UPDATE OF clause of a trigger definition. Correct the ALTER statement to specify unique names for each of the ALTER COLUMN clauses.

This error can also occur on CREATE TABLE when a column list of a PRIMARY KEY, FOREIGN KEY, or UNIQUE clause contains two or more occurrences of the same column name.

DSNH628I **E** *csectname* **LINE** *nnnn* **COL** *cc* **MUTUALLY EXCLUSIVE CLAUSES HAVE BEEN SPECIFIED.**

Explanation: Mutually exclusive clauses were specified in one or more of the following ways:

- A CREATE TABLESPACE statement contains both the SEGSIZE and NUMPARTS clauses
- A CREATE TABLESPACE statement contains both the SEGSIZE and LARGE clauses.
- A CREATE TABLESPACE statement contains both the SEGSIZE and MEMBER CLUSTER clauses.
- A CREATE or ALTER TABLESPACE contains both the LOCKPART YES and LOCKSIZE TABLESPACE
- A 'column-definition' contains both NOT NULL and DEFAULT NULL clauses.
- A 'column-definition' contains both FIELDPROC and DEFAULT clauses.
- A select-statement contains both the update-clause and the FOR FETCH ONLY clause.
- An ALTER TABLE statement contains both a DROP CONSTRAINT clause and either a DROP FOREIGN KEY clause or a DROP CHECK clause.
- A CREATE or ALTER TABLESPACE contains both the LOCKPART YES and LOCKSIZE TABLESPACE
- A CREATE TRIGGER statement specifies more than one correlation name for OLD, NEW, OLD_TABLE, or NEW_TABLE. Each of these correlation specific can appear no more than once in the CREATE TRIGGER statement.
- A CREATE FUNCTION statement contains both a CAST FROM clause and a SOURCE clause.
- A CREATE FUNCTION statement contains both a SOURCE clause and a RETURNS TABLE clause.
- A CREATE FUNCTION statement contains both a SOURCE clause and a clause used to define an external function (EXTERNAL, LANGUAGE, NO SQL).
- A CREATE or ALTER PROCEDURE statement attempts to use the NO WLM ENVIRONMENT and PROGRAM TYPE SUB options. When NO WLM ENVIRONMENT is used, then PROGRAM TYPE MAIN must also be used.
- A CREATE or ALTER PROCEDURE statement attempts to use both NO WLM ENVIRONMENT and either USER or DEFINER for EXTERNAL SECURITY. When NO WLM ENVIRONMENT is used, then PARAMETER STYLE must be either GENERAL or SIMPLE CALL WITH NULLS.
- A CREATE or ALTER PROCEDURE statement contains both a LANGUAGE REXX clause, and a PARAMETER STYLE DB2SQL or PARAMETER STYLE JAVA clause.
- An ALTER TABLE statement contains both an ALTER COLUMN clause and a VALIDPROC clause.
- An ALTER TABLE statement contains both an ALTER COLUMN clause and a clause other than the check constraint clause.

Severity: 8 (error)

System action: The statement was not executed.

Programmer response: Change the options specified in the statement and reissue the statement.

DSNH630E **E** *csectname* **LINE** *nnnn* **COL** *cc* **THE WHERE NOT NULL SPECIFICATION IS INVALID FOR TYPE 1 INDEXES**

Explanation: Type 1 indexes cannot be created with the WHERE NOT NULL specification.

Severity: 8 (error)

System action: The statement cannot be executed.

Programmer response: Either the index must be created as a type 2 index, or the WHERE NOT NULL specification must not be used.

DSNH637I **E** *csectname* **LINE** *nnnn* **COL** *cc* **DUPLICATE** *keyword* **KEYWORD**

Explanation: The identified keyword was specified twice in the statement.

Severity: 8 (error)

System action: If the containing SQL statement is a declaration, it is discarded. If the containing SQL statement is executable, it is replaced by host language statements that will, if executed, set an error code in the SQLCA.

User response: Remove the duplicate specification.

DSNH644I **E** *csectname* **INVALID VALUE SPECIFIED FOR KEYWORD** *keyword* **IN** *stmt-type* **STATEMENT**

Explanation: The value specified for the *keyword* parameter in the *stmt-type* SQL statement is not a permitted value.

Severity: 8 (error)

System action: If the SQL statement containing the problem is a declaration, it is discarded. If the SQL statement containing the problem is executable, it is replaced by host language statements that will, if executed, set an error code in the SQLCA.

User response: Correct the statement. Refer to Chapter 5 of *DB2 SQL Reference* for information about the allowable values for the *keyword* in the *stmt-type* statement.

DSNH683I **E** *csectname* **LINE** *nnnn* **COL** *cc***. THE SPECIFICATION FOR COLUMN, DISTINCT TYPE, FUNCTION, OR PROCEDURE** *data-item* **CONTAINS INCOMPATIBLE CLAUSES**

Explanation: There is an error in the data item specification in a statement. Incompatible specifications are present such as "INTEGER and FOR BIT DATA", or "INTEGER AS LOCATOR". The location of the error is given by *data-item* as follows:

- For a CREATE or ALTER TABLE statement, *data-item* gives the name of the column containing the error. The error could be an invalid specification of FOR BIT DATA, FOR SBCS DATA, FOR MIXED DATA, or FIELDPROC for column data-item.
- For a CREATE FUNCTION or CREATE PROCEDURE statement, *data-item* is a token that identifies the area of the problem in the statement. For example, "PARAMETER 3", "RETURNS", or "CAST FROM".
- For a CREATE DISTINCT TYPE statement, *data-item* gives the name of the type being defined
- Otherwise, *data-item* is a token that identifies the area of the problem in a parameter list for a function. For example, "PARAMETER 5".

Severity: 8 (error)

System action: The statement cannot be processed.

User response: Correct and reissue the failing statement.

DSNH684I **E** *csectname* **LINE** *nnnn* **COL** *cc* **LENGTH OF LITERAL LIST BEGINNING** *string* **IS TOO LONG**

Explanation: The length of the literal list beginning with *string*, excluding insignificant blanks and delimiting parenthesis, is greater than 255.

Severity: 8 (error)

System action: The SQL statement cannot be executed.

System programmer response: Correct the SQL statement. Refer to Chapter 2 of *DB2 SQL Reference* for information about writing literal lists.

DSNH700I **I** *csectname* **THE FIPS STANDARD IS USED FOR FLAGGING**

Explanation: This message identifies the standard used for flagging during execution.

Severity: 0 (informational)

System action: Processing continues.

DSNH701I **I** *csectname* **THE IBM SQL STANDARD IS USED FOR FLAGGING**

Explanation: This message identifies the standard used for flagging during execution.

Severity: 0 (informational)

System action: Processing continues.

DSNH702I **W** *csectname* **LINE** *nnnn* **COL** *cc* **THE STATEMENT DEVIATES FROM THE SQL STANDARD AT THE TOKEN COMMENCING WITH THE TEXT** *text*

Explanation: A syntax deviation from the standard named in the earlier message (FIPS or IBM SQL) was found in the SQL statement.

Severity: 4

User response: Check and correct the SQL statement.

DSNH703I **W** *csectname* **LINE** *nnnn* **COL** *cc***. A DATA DEFINITION STATEMENT IS NOT WITHIN A CREATE SCHEMA STATEMENT**

Explanation: The FIPS standard requires all data definition statements to be contained within a CREATE SCHEMA statement.

Severity: 4

User response: Check and correct the SQL statement.

DSNH704I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THERE IS A NONSTANDARD EMBEDDED COMMENT**

Explanation: A comment embedded within an SQL statement does not meet the requirements of the standard being flagged: The FIPS standard is being checked and the comment does not begin with at least two consecutive hyphens.

Severity: 4

User response: Check and correct the SQL statement.

DSNH705I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **AN SQL SYNTAX DEVIATION OCCURRED. THE STATEMENT IS NOT COMPLETE**

Explanation: The SQL statement terminated before all required elements were found.

Severity: 4

User response: Check and correct the SQL statement.

DSNH706I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE IDENTIFIER** *token* **IS GREATER THAN 18 CHARACTERS**

Explanation: Authorization identifier, table identifier, column name, correlation name, module name, cursor name, procedure name, or parameter name has a length of more than 18 characters.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH707I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE COLUMN** *column* **HAS AN INVALID LENGTH, PRECISION, OR SCALE ATTRIBUTE**

Explanation: One of the following conditions was not met:

- The length value should be greater than zero.
- The precision value should be greater than zero.
- The scale value should not be greater than precision.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH708I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE DATA TYPE OF** *token* **SHOULD BE EXACT NUMERIC WITH A SCALE OF ZERO**

Explanation: The data type of INDICATOR PARAMETER or INDICATOR VARIABLE should be exact numeric with a scale of zero.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH709I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE SET FUNCTION REFERENCES COLUMN column**

Explanation: One of the following conditions was not met:

- A column specification of a DISTINCT set function cannot reference a column derived from a set function specification.
- A column specification of an ALL set function cannot reference a column derived from a set function specification.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH710I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **A VALUE EXPRESSION CONTAINING** *column* **CANNOT INCLUDE OPERATORS**

Explanation: A value expression cannot include operators in an outer reference column specification.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH711I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE COLUMN SPECIFICATION IS MISSING OR INVALID FOR THE ALL SET FUNCTION** *function*

Explanation: The value expression of an ALL set function specification should include a column specification.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH712I **W** *csectname* **LINE** *nnnn* **COL** *cc* **THE COLUMN** *column* **IS NOT UNIQUE AND REQUIRES QUALIFICATION.**

Explanation: The identified column is duplicated within the current scope. Qualification must be given to uniquely identify the required column.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH713I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **A VALUE EXPRESSION MAY NOT CONTAIN A SET FUNCTION SPECIFICATION**

Explanation: The value expression of an ALL set function might not contain a set function specification.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH714I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE COLUMN** *column* **DOES NOT IDENTIFY A COLUMN OF A TABLE WITHIN THE CURRENT SCOPE**

Explanation: One of the following conditions was found:

- The table or correlation name used as a qualifier does not exist.
- The column name does not exist within the current scope or the scope of the qualifier.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH715I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **A SET FUNCTION THAT CONTAINS THE OUTER REFERENCED COLUMN** *column* **IS NOT IN THE SUBQUERY OF A HAVING CLAUSE**

Explanation: A set function specification must be contained in a subquery of a HAVING clause if it contains an outer referenced column specification.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH716I **W** *csectname* **LINE** *nnnn* **COL** *cc* **THE RESULT OF A SUM OR AVG FUNCTION CANNOT BE A CHARACTER STRING**

Explanation: The character string is not valid for the result of a SUM or AVG set function specification.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH717I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE OPERATOR** *operator* **IS INVALID IN THIS CONTEXT**

Explanation: One of the following conditions was not met:

- A value expression containing a distinct set function cannot include dyadic operators.
- The first character following a monadic operator cannot be a plus or minus sign.
- The value expression should not contain operators if primary is a character string.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH718I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE** *exptype* **EXPRESSION IS COMPARING INCOMPATIBLE DATA TYPES**

Explanation: The data types of one of the following (identified by *exptype*) do not match:

exptype Meaning

COMPARISON
Comparison operators must match.

BETWEEN
The three value expressions must match.

IN
Value expression, subquery, and all value specifications must match.

QUANTIFIED
Value expression and subquery must match.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH719I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE OPERAND IN THE LIKE PREDICATE IS NOT A CHARACTER STRING**

Explanation: One of the following conditions was found:

- The data type of the column in a LIKE predicate is not a character string.
- The data type of the pattern in a LIKE predicate is not a character string.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH720I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE ESCAPE CHARACTER MUST BE A ONE-BYTE CHARACTER STRING**

Explanation: The escape character in a LIKE predicate must have the data type of character string with a length of 1.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH721I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE WHERE CLAUSE, GROUP BY CLAUSE, OR HAVING CLAUSE IS INVALID FOR THE GROUPED VIEW** *table*.

Explanation: If the table identified in the FROM clause is a grouped view, then the table expression must not contain a WHERE clause, GROUP BY clause, or HAVING clause.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH722I **W** *csectname* **LINE** *nnnn* **COL** *cc*. *name* **OCCURS MORE THAN ONCE IN A FROM CLAUSE**

Explanation: One of the following conditions was found:

- The table name occurs more than once in a FROM clause.
- The correlation name is identical to a table name or to another correlation name in a FROM clause.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH723I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **ONLY ONE TABLE REFERENCE IS ALLOWED IN THE FROM CLAUSE FOR A GROUPED VIEW**

Explanation: If the table identified by table name is a grouped view, then the FROM clause must contain exactly one table reference.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH724I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **A REFERENCE TO THE COLUMN** *column* **WHICH IS DERIVED FROM A SET FUNCTION IS INVALID IN A WHERE CLAUSE**

Explanation: A value expression directly contained in the search condition of a WHERE clause must not include a reference to a column derived from a set function specification.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH725I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **A WHERE CLAUSE MUST BE CONTAINED IN A HAVING CLAUSE WHEN A FUNCTION IS USED**

Explanation: If a value expression directly contained in the search condition is a set function specification, then the WHERE clause must be contained in a HAVING clause.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH726I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE COLUMN SPECIFICATION FOR** *column* **MUST BE AN OUTER REFERENCE**

Explanation: If a value expression directly contained in the search condition is a function, then the column specification in the function must be an outer reference.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH727I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE COLUMN** *column* **IS DUPLICATED WITHIN THE CURRENT SCOPE**

Explanation: The specified column is duplicated within the current scope.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH728I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE COLUMN SPECIFICATION FOR table SHOULD BE A GROUPING COLUMN OR SHOULD BE SPECIFIED WITHIN A SET FUNCTION**

Explanation: Each column specification contained in a subquery in the search condition of a HAVING clause must reference a grouping column or be specified within a set function specification.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH729I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE DEGREE OF TABLE** *table* **SHOULD BE 1 WHEN USING A RESULT SPECIFICATION OF ***

Explanation: The degree of the table expression must be 1 if the result specification '*' is specified in a

subquery of any predicate other than an EXISTS predicate.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH730I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE SET FUNCTION IS NOT VALID FOR THE TABLE EXPRESSION BEGINNING WITH** *table*

Explanation: One of the following conditions was not met:

- If the table expression is a grouped view, the result specification of a subquery cannot contain a set function specification.
- If the table expression is a grouped view, the select list of a query specification cannot contain a function.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH731I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE COLUMN SPECIFICATION FOR** *column* **IS INVALID**

Explanation: One of the following conditions was not met:

- For a grouped table, the column specification must reference a grouping column or be specified within a set function specification.
- If it is not a grouped table, and the value expression includes a set function specification, then each column specification must be specified within a set function.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH732I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **DISTINCT IS SPECIFIED MORE THAN ONCE.**

Explanation: One of the following conditions has not been met:

- DISTINCT must not be specified more than once in a query specification, excluding any subquery of that query specification.
- DISTINCT must not be specified more than once in a subquery, excluding any subquery contained in that subquery.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH733I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE COMPARISON PREDICATE SUBQUERY CANNOT CONTAIN A GROUP BY OR HAVING CLAUSE**

Explanation: If a subquery is specified in a comparison predicate, then the table expression whose FROM clause identifies the named table must not contain a GROUP BY clause or a HAVING clause.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH734I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE COMPARISON PREDICATE SUBQUERY CANNOT IDENTIFY A GROUPED VIEW**

Explanation: If a subquery is specified in a comparison predicate, then the table expression whose FROM clause identifies the named table must not identify a grouped view.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH735I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **INVALID AUTHORIZATION IDENTIFIER** *authid*

Explanation: The authorization identifier prefixed to the table name is invalid.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH736I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE TABLE OR VIEW** *name* **ALREADY EXISTS**

Explanation: The table name or view name specified already exists in the catalog.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH737I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE COLUMN DEFINITION IS MISSING**

Explanation: At least one column definition should be specified for a CREATE TABLE.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH738I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE DATA TYPE** *type1* **OF THE TARGET IS NOT COMPATIBLE WITH THE DATA TYPE** *type2* **OF THE SOURCE**

Explanation: Data types should match between source and target in a FETCH and SELECT statement.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH739I **E** *csectname* **LINE** *nnnn* **COL** *cc*. **CREATE FUNCTION** *function-name* **FAILED BECAUSE FUNCTIONS CANNOT MODIFY DATA WHEN THEY ARE PROCESSED IN PARALLEL.**

Explanation: The function cannot be created because ALLOW PARALLEL and MODIFIES SQL DATA were both specified explicitly or implicitly. A function cannot be parallelized if it modifies data.

Severity: 8 (error)

System action: The statement cannot be executed.

User response: Specify DISALLOW PARALLEL or change the MODIFIES SQL DATA to NO SQL, CONTAINS SQL or READS SQL DATA.

DSNH740I *xxxxxx phase* **FINISHED, RC =** *retcode*, **LISTING IN** *data-set*

Explanation: The DSNH INST phase completed with return code *retcode*. The *phase* is one of the following:

PLI Macro Pass
Precompiler
CICS Command Translator
Bind Package
Bind Plan
Compile/assembly
PLI
Prelink (unique to C/370 Compiler)
Link
Run

If the CLIST puts the listing into a data set, the name of the data set is shown as *data-set*. If the PRINT parameter is LEAVE, NONE, or TERM, the 'LISTING IN *data-set*' part of this message is omitted.

Refer to message DSNH756I for an explanation of the print data set names assigned during processing.

This message is issued by the following CLIST: DSNH

Severity: *retcode*

System action: Processing continues as requested.

User response: This phase is complete. If there are errors, other messages are produced. The next phase requested begins. The order of the phases is:

1. PLI Macro Pass
2. Precompiler
3. CICS Command Translator
4. Bind Package
5. Bind Plan
6. ASM, ASMH, C (C/370 Compiler), COBOL, COBOL2, FORTRAN, PLI
7. Prelink
8. Link
9. Run

DSNH741I *process1***(NO) IMPLIES THAT** *process2* **CANNOT BE RUN**

Explanation: An inconsistent processing set was specified. An earlier phase and a later phase were requested, but not the intermediate step needed to process data between them. For example, requesting the precompile and the link edit without the compile produces this error. Values for *process1* are PRECOMP, CICSXLAT, COMPILE, LINK, or BIND. Values for *process2* are COMPILE, LINK, or RUN.

This message is issued by the following CLIST: DSNH

Severity: 4 (warning)

System action: *process2* is not performed, but other processing is done.

User response: Specify a consistent combination of processing. Application program preparation needs to run all the steps to provide a correct application program. The process does allow beginning or ending steps to be run at a different time. See message DSNH740I for the required step order.

DSNH750I **INVALID DATA SET:** *clist-keyword clist-keyword-parameter*

Explanation: The data set name specified on the *clist-keyword-parameter* contains unbalanced parenthesis.

This message is issued by the following CLIST: DSNH

Severity: 8 (error)

System action: The CLIST is terminated.

User response: If a member name is to be associated with the specified data set, the member name must be enclosed in parentheses.

DSNH751I *clist* **ERROR IN** *parameter* **VALUE** *value*

Explanation: The specified CLIST keyword *parameter* does not have an acceptable value.

Severity: 8 (error)

System action: The CLIST processing is terminated.

User response: Refer to the DSNH (TSO clist) section in *DB2 Command Reference* for a complete list of the supported CLIST parameters and their associated

values. Correct the invalid parameter, and resubmit the job.

DSNH752I INPUT DATA SET NAME IS MISSING OR NONE

Explanation: The DSNH CLIST requires a value for the input data set name (INPUT). You cannot use an input data set name of NONE.

This message is issued by the following CLIST: DSNH

Severity: 0 (successful)

System action: The CLIST prompts for a name, if prompting is allowed. If prompting is not allowed, for example in batch or when PROFILE NOPROMPT was entered, the CLIST terminates.

User response: Specify an INPUT name. See message DSNH753I.

DSNH753I ENTER INPUT DATA SET NAME

Explanation: The CLIST is requesting the name of the input data set.

This message is issued by the following CLIST: DSNH

Severity: 0 (successful)

System action: The system waits for user to enter data set name or for the CLIST to be terminated.

User response: Enter the name of the data set. Do not specify the parameter or command, but just enter the name of the data set. The standard TSO prefixing will be done if the data set name is not entered within apostrophes.

DSNH754I *clist* **OUTNAME PARAMETER** *name* **MUST NOT CONTAIN SPECIAL CHARACTERS**

Explanation: The DSNH and DSNHC CLISTs, *clist*, use the OUTNAME parameter to build names for a number of data sets that are passed from phase to phase. These names are prefixed by an authorization ID or TSO prefix value and have a number of suffixes. For this reason, OUTNAME parameter *name* must not contain special characters. It can contain a period or national characters. It cannot have apostrophes to specify a single data set name.

Severity: 8 (error)

System action: The CLIST is terminated.

User response: Correct the input request. The default for this parameter is TEMP, which should be adequate for most usage.

DSNH755I OUTNAME PARAMETER *name* **MUST NOT BEGIN THE SAME AS INPUT** *input*

Explanation: The DSNH CLIST uses the OUTNAME parameter *name* to build a number of data sets that are passed from phase to phase. These data sets could conflict with the INPUT data set (*input*) if the names are the same, causing the INPUT data set to be overwritten. To avoid this problem, the names are not allowed to have the same initial characters.

This message is issued by the following CLIST: DSNH

Severity: 8 (error)

System action: The CLIST is terminated.

User response: Correct the input request. Specify a different OUTNAME. The default for OUTNAME is TEMP.

DSNH756I *clist* **PRINT PARAMETER** *name* **MUST NOT CONTAIN SPECIAL CHARACTERS**

Explanation: The DSNH and DSNHC CLISTs (*clist*) use the PRINT parameter to build names for data sets used for the print listings from various phases. These names are prefixed by an authorization ID or a TSO prefix value and have a number of suffixes. The PRINT parameter *name* can contain a period or national characters, but it cannot contain special characters. It cannot have apostrophes to specify a single data set name.

The data set names built using the PRINT(print) qualifier are the following:

print.CXLIST
: CICS Command Translator

print.LINKLIST
: Linkage Editor

print.LIST
: Compiler or Assembler

print.PCLIST
: DB2 Precompiler

print.SYSCPRT.LIST
: C/370 Compiler

print.SYSOUT.PRELLIST
: C/370 Prelink Utility

Severity: 8 (error)

System action: The CLIST terminates.

User response: Correct the *name* specified with the PRINT parameter.

DSNH757I ERROR IN PLAN NAME LENGTH *value*

Explanation: The *value* is too long for a plan name. A plan name has a maximum of 8 characters.

This message is issued by the following CLIST: DSNH

Severity: 8 (error)

System action: The CLIST is terminated.

User response: Specify a plan name with 8 or fewer characters.

DSNH758I HOST(*host-language*) IS NOT PERMITTED WITH RUN(IMS) OR RUN(CICS)

Explanation: There is no facility for connecting a FORTRAN program to either IMS or CICS.

This message is issued by the following CLIST: DSNH

Severity: 8 (error)

System action: The CLIST terminates.

User response: Use the specified language in the TSO or batch environment.

DSNH759I *clist* **TERMINATING,** *phase* **RC =** *retcode*

Explanation: The DSNH or DSNHC *clist* terminated at *phase* because the return code *retcode* was greater than or equal to the RCTERM parameter value.

Severity: *retcode*

System action: The CLIST terminates.

User response: Determine the cause for the high return code. If the parameters include TERM(NONE) and direct the PRINT to a data set, you might need to search the print data sets for the listing of the error. Refer to message DSNH756I for an explanation of the print data set names assigned during processing.

Problem determination: The location of the messages describing the prior errors depends upon the values of the TERM and PRINT parameters. The messages can be directed to the terminal, to preallocated data sets, or to data sets dynamically allocated during CLIST processing.

If TERM(NONE) PRINT(NONE) was specified, the messages are not available. Execute the CLIST again and specify a location for the error messages.

To find the documentation, note the immediately preceding DSNH740I message for the location of the listings. If the 'LISTING IN data-set' portion of the message is missing, specify a PRINT data set, SOURCE(YES) and XREF(YES), and execute the CLIST again. Use TERM(TERM) to direct error messages to your terminal. These listings and execution of the DSNH CLIST are required for problem diagnosis.

If the error is in the CLIST or the error location is CLIST CHECKING, use CONTROL(SYMLIST) to produce a detailed trace of the CLIST execution and provide the trace listing.

Check for DSNH751I messages that describe the error.

DSNH760I PRELINK OPTION INVALID WITH HOST LANGUAGE (*host-language*)

Explanation: Invocation of the PRELINK step is not supported by *host-language*. This option is only valid for the following DSNH HOST language compilers: C, CPP, and IBMCOB.

This message is issued by the following CLIST: DSNH

Severity: 4 (error)

System action: The DSNH PRELINK parameter is ignored and prelink utility processing is bypassed. Clist processing continues normally with a return code of 4.

User response: Refer to the description of the DSNH PRELINK keyword option in *DB2 Command Reference*.

DSNH770I W *csectname* **LINE** *nnnn* **COL** *cc*. **YOU MUST DECLARE COLUMN** *column* **AS NOT NULL**

Explanation: All columns identified as UNIQUE must be defined with the NOT NULL option.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH771I W *csectname* **LINE** *nnnn* **COL** *cc*. **THE VIEWED TABLE** *table* **MUST BE UPDATEABLE**

Explanation: When the WITH CHECK OPTION is specified, then the viewed table must be updateable.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH772I W *csectname* **LINE** *nnnn* **COL** *cc*. **INVALID NUMBER OF COLUMN NAMES**

Explanation: The number of column names in the view column list should be the same as the degree of the table specified by the query specification.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH773I W *csectname* **LINE** *nnnn* **COL** *cc*. **YOU MUST DECLARE THE CURSOR** *cursor* **BEFORE USE**

Explanation: The cursor specified was not declared in a DECLARE CURSOR statement.

Severity: 4

User response: Note the deviation from the applicable

standard. Check and correct the SQL statement.

DSNH774I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE CURSOR** *cursor* **IS ALREADY DECLARED**

Explanation: The cursor specified was already declared in a DECLARE CURSOR statement.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH775I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **ONLY '*' OR COLUMN NAMES ARE VALID IN THIS CONTEXT**

Explanation: When UNION is specified, the select list for the specification of the two table expressions identified by the query expression and query term should consist of '*' or column specifications.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH776I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE DESCRIPTIONS OF TABLES IDENTIFIED BY THE QUERY EXPRESSION BEGINNING WITH TABLE:** *table1* **AND THE QUERY TERM BEGINNING WITH TABLE:** *table2* **SHOULD BE IDENTICAL**

Explanation: When UNION is specified, the descriptions of the two tables should be identical, except for the column names.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH777I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE SORT SPECIFICATION** *number* **IS OUTSIDE THE DEGREE OF TABLE** *table*

Explanation: If the sort specification contains an unsigned integer, it should be greater than zero and not greater than the number of columns in the table.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH778I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE TABLE** *table* **IS A READ-ONLY TABLE**

Explanation: A DELETE, an INSERT, or an UPDATE statement was specified on a read-only table.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH779I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE TABLE** *table* **MUST NOT BE IDENTIFIED IN A FROM CLAUSE OF ANY SUBQUERY CONTAINED IN A SEARCH CONDITION**

Explanation: A table specified in a DELETE or an UPDATE statement cannot be used in a FROM clause of any subquery contained in its search condition.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH780I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE TABLE** *table1* **IS NOT THE FIRST TABLE** *table2* **SPECIFIED IN THE DECLARE CURSOR STATEMENT**

Explanation: The table specified in a DELETE or an UPDATE statement should be the first table specified in the FROM clause of the DECLARE CURSOR statement.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH781I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE NUMBER OF TARGET SPECIFICATIONS DO NOT MATCH THE DEGREE OF TABLE** *table*

Explanation: The number of target specifications in the FETCH statement do not match the degree of the table specified.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH782I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE TARGET TABLE** *table* **OF THE INSERT STATEMENT IS ALSO IN THE FROM CLAUSE OR IN A CONTAINED SUBQUERY**

Explanation: The named table must not be identified in a FROM clause of a query specification or of any subquery contained in the query specification.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH783I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE NUMBER OF COLUMNS SPECIFIED DOES NOT MATCH THE NUMBER OF VALUES SPECIFIED**

Explanation: In an INSERT statement, the number of columns specified does not match the number of values specified.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH784I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE NUMBER OF COLUMNS SPECIFIED DOES NOT MATCH THE DEGREE OF TABLE** *table* **IN THE QUERY SPECIFICATION**

Explanation: The number of columns specified in an INSERT statement does not match the degree of the table specified in the query specification.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH785I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THERE IS A DATA TYPE OR LENGTH MISMATCH BETWEEN COLUMN** *column* **AND ITS INSERT OR UPDATE ITEM**

Explanation: One of the following conditions was not met:

- If the data type of the column name is character string, then the corresponding item in the INSERT or UPDATE statement should be character string with a length equal to or less than the length of the column name.
- If the data type of the column name is exact numeric, then the corresponding item in the INSERT or UPDATE statement should be exact numeric.
- If the data type of the column name is approximate numeric, then the corresponding item in the INSERT or UPDATE statement should be approximate numeric or exact numeric.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH786I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **YOU CANNOT USE A GROUP BY OR HAVING CLAUSE OR IDENTIFY A GROUPED VIEW IN THIS CONTEXT**

Explanation: The table identified in the FROM clause of the table expression in a SELECT statement should not include a GROUP BY or HAVING clause and should not identify a grouped view.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH787I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE NUMBER OF ELEMENTS SPECIFIED IN THE SELECT LIST SHOULD MATCH THE NUMBER IN THE TARGET LIST**

Explanation: In the SELECT statement, the number of elements specified in the select list should match the number of elements in the target list.

Severity: 4 (warning)

System action: The statement cannot be processed.

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH788I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE SET FUNCTION** *function* **IS NOT ALLOWED IN AN UPDATE STATEMENT**

Explanation: A value expression in a SET clause of an UPDATE statement should not include a set function specification.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH789I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **YOU CANNOT SPECIFY NULL FOR NOT NULL COLUMN** *column*

Explanation: If NULL is specified in the SET clause of the UPDATE statement, then the corresponding column should allow nulls.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH790I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THERE IS AN INVALID SQL TERMINATOR FOR THE HOST LANGUAGE**

Explanation: The SQL statement was not terminated

by the correct construct for the host language being processed.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH791I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **YOU MUST DEFINE THE HOST VARIABLE** *identifier*

Explanation: The host variable named was either not defined within the program or defined outside the SQL declaration section.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH792I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE TABLE** *table* **WAS NOT FOUND IN THE CATALOG**

Explanation: The named table does not exist in the catalog.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH793I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THERE IS AN INVALID IDENTIFIER FORMAT** *identifier* **FOR HOST LANGUAGE** *language*

Explanation: The format of the identifier name displayed does not meet the requirements of the host language.

Severity: 4

User response: Note the deviation from the applicable standard. Check and correct the SQL statement.

DSNH794I **W** *csectname* **LINE** *nnnn* **COL** *cc*. **THE OBJECT** *object-name* **CANNOT BE CREATED BECAUSE THE FIRST THREE CHARACTERS ARE RESERVED FOR SYSTEM OBJECTS**

Explanation: In general, SYS is a reserved prefix for names. The only exceptions to this rule include the following conditions:

- SYSADM and SYSTOOLS are valid schema names.
- SYSPROC is a valid schema name for stored procedures.
- SYSIBM is a valid schema name for a stored procedure when the current application process has the SYSADM or SYSCTRL privilege.
- SYSTOOLS is a valid schema name when the current application process has the SYSADM or SYSCTRL privilege.

This message is also issued if an attempt is made to grant the CREATEIN, ALTERIN or DROPIN privileges on a schema with the SYS prefix. The same exceptions apply to the grant.

This condition is similar to the condition reported in SQLCODE -20074.

Severity: 4

User response: Select a name that does not start with a reserved prefix.

DSNH867I **E** *csectname* **LINE** *nnnn* **COL** *cc*. **INVALID SPECIFICATION OF A ROWID COLUMN**

Explanation: For a CREATE TABLE statement, the specification of a ROWID column might be invalid for one of the following reasons:

- A ROWID column cannot be a column of a temporary table.
- A ROWID column cannot be a column of a table with an EDITPROC.
- A ROWID column cannot be a column of a foreign key.
- A ROWID column cannot be a column of a primary key.
- A ROWID column cannot be referenced in the check-condition of a table check constraint.

Severity: 8

System action: The statement was not executed.

User response: Correct the syntax.

DSNH870I **W** *csectname* **THERE IS NOT ENOUGH MEMORY FOR THE FLAGGER; FLAGGING IS DISCONTINUED**

Explanation: The flagger could not allocate enough storage to load all required modules.

Severity: 4

System action: Precompiling continues but flagger operation is discontinued.

User response: Increase the region size in the JCL for the application. Allocate more storage for flagger operation.

DSNH871I **W** *csectname* **AN ERROR OCCURRED WHEN ATTEMPTING TO LOAD THE FLAGGER MODULE** *module*. **FLAGGING IS DISCONTINUED**

Explanation: A flagger module was not found during flagger initialization.

Severity: 4

System action: Precompiling continues but flagger operation is discontinued.

User response: Refer to the message produced by the operating system for details of causes and proceed as instructed for that message.

DSNH872I W *csectname* **AN ERROR OCCURRED WHEN INITIALIZING THE FLAGGER. FLAGGING IS DISCONTINUED**

Explanation: An internal error occurred during flagger initialization.

Severity: 4

System action: Precompiling continues but flagger operation is discontinued.

User response: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNH873I W *csectname* **THE FLAGGER COULD NOT PROCESS THE SQL STATEMENT. FLAGGING IS DISCONTINUED**

Explanation: An SQL statement passed to the flagger caused an internal error within the flagger.

Severity: 4

System action: Precompiling continues but flagger operation is discontinued.

User response: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNH874I W *csectname* **AN ERROR OCCURRED WHEN TERMINATING THE FLAGGER**

Explanation: An internal error occurred during flagger termination.

Severity: 4

System action: Precompiling continues but flagger operation is discontinued.

User response: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNH875I W *csectname* **AN ERROR OCCURRED WHEN ATTACHING TO DB2, REASON CODE: X'***reason***'; FLAGGING CONTINUED AS SYNTAX ONLY**

Explanation: An error occurred while trying to connect or disconnect from the DB2 subsystem.

Severity: 4

System action: Processing continues without semantics flagging.

User response: Refer to the reason code returned in this message and perform corrective actions.

DSNH876I W *csectname* **AN ERROR SQLCODE** *sqlcode* **WAS RETURNED DURING FLAGGER ACCESS TO THE CATALOG OBJECT** *authid.table***. MODULE NAME:** *module* **ERROR REASON CODE: X'***reason***'; FLAGGING CONTINUES WITH SYNTAX CHECK ONLY**

Explanation: An error occurred during the execution of a flagger access to the DB2 catalog.

Severity: 4

System action: Processing continues without semantics flagging.

User response: Use the provided SQL return code, object type, and name to identify the problem. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNH877I W *csectname* **NOT ENOUGH MEMORY FOR FLAGGER OPERATION: RETURN CODE: return**

Explanation: Insufficient storage is available for flagger operation.

Severity: 4

System action: Processing continues without flagging.

User response: Check whether the storage allocation for the precompiler can be increased. Allocate more storage for flagger operation.

DSNH878I W *csectname* **ERROR RELEASING FLAGGER MEMORY; FLAGGING IS DISCONTINUED**

Explanation: An error occurred when a flagger module attempted to release storage back to the flagger pool.

Severity: 4

System action: Processing continues without flagging.

User response: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNH879I **W** *csectname* **ERROR RELEASING FLAGGER MEMORY; RETURN CODE** *return*

Explanation: An error occurred when a flagger module attempted to release storage back to the operating system.

Severity: 4

System action: Processing continues without flagging.

User response: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNH880I **W** *csectname* **AN ERROR WAS RETURNED DURING FLAGGER ACCESS TO THE DB2 CATALOG; REASON CODE: X'***reason***' RETURN CODE X'***return***'**

Explanation: An error occurred when a flagger module attempted to access the system catalog.

Severity: 4

System action: Processing continues without flagging.

User response: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNH901I **S** *csectname* **U LINE** *nnnn* **COL** *cc* **INTERNAL PRECOMPILER ERROR – REASON CODE** *ccc*

Explanation: An internal compiler consistency check failed. Reason codes are:

000 The message formatter was called with an unlisted internal message number.

001 The parser received a token code of 0 from the lexical scan. Lexical scan refers to a scan of words rather than syntax.

002 Error in storage suballocation control blocks.

- The error might have occurred because a DB2 SQL statement limitation was exceeded. The SQL statement might be referencing, creating, or declaring more columns than the DB2 defined maximum number allowed for an SQL statement. Another possibility is that too many host variables are being referenced in the SQL statement.

003 Error in a storage suballocation request.

- This error might have occurred because of storage constraints when building the DBRM entry for the SQL statement or because the SQL statement exceeds the DB2 maximum statement length.

004 Error in pointer structures.

005 On BIND or REBIND, a host variable was not found.

006 Unexpected value for parser option.

007 The size of a DBRM entry exceeds the
maximum size.

008 Internal conversion error.

- An error occurred when converting a timestamp version-id.

Severity: 16 (unrecoverable error)

System action: The precompiler is terminated immediately.

User response: Notify the system programmer.

System programmer response: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: A copy of the application program source and the printout is useful (specify SOURCE and XREF).

DSNH902I **S** *csectname* **LINE** *nnnn* **COL** *cc* **SEVERE INTERNAL PRECOMPILER ERROR IN CONVERSION, REASON** *reason-code*

Explanation: DB2 parses SQL statements in Unicode UTF-8, which is also known as CCSID 1208. DB2 will convert the application program internally (not externally) to UTF-8 for processing if an application program uses a different CCSID (for example, an EBCDIC CCSID). After processing, DB2 converts the output back to the original CCSID.

This message indicates that a severe error has occurred in DB2's internal conversion. The reason code indicates the context of this error:

000 Substitution occurred on conversion from the source CCSID to CCSID 1208.

001 The internal buffer is too small for conversion from the source CCSID to CCSID 1208.

002 A miscellaneous error occurred on conversion from the source CCSID to CCSID 1208.

003 No conversion is defined from the source CCSID to CCSID 1208.

004 A miscellaneous error occurred on conversion from the source CCSID to CCSID 1208.

005 Substitution occurred on conversion from CCSID 1208 to the source CCSID.

006 The internal buffer is too small for conversion from CCSID 1208 to the source CCSID.

007 An invalid code point was found during conversion from CCSID 1208 to the source CCSID.

008 Another type of invalid input character was found during conversion from CCSID 1208 to the source CCSID.

009 A miscellaneous error occurred during conversion from CCSID 1208 to the source CCSID.

010 No conversion is defined from CCSID 1208 to the source CCSID.

011 A miscellaneous error occurred during conversion from CCSID 1208 to the source CCSID.

Severity: 16 (unrecoverable error)

System action: The precompiler is terminated immediately.

User response: Notify the system programmer.

System programmer response: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: A copy of the application program source and the print-out is useful (specify SOURCE and XREF).

DSNH904I *csectname* **U STORAGE OVERFLOW**

Explanation: More storage is needed, but could not be obtained during precompiler/parser execution.

Severity: 16 (unrecoverable error)

System action: The precompiler/parser is terminated immediately.

System programmer response: The storage required for the current statement could not be acquired or exceeds the maximum for a single request. If the region size is being used completely, check whether the region can be expanded. If it cannot be expanded, the SQL statement is too large to be processed.

DSNH1760I E *csectname* **LINE** *nnnn* **COL** *cc*. **CREATE STATEMENT FOR STORED PROCEDURE** *procedure-name* **MUST HAVE A VALID LANGUAGE CLAUSE AND EXTERNAL CLAUSE.**

Explanation: The CREATE statement for procedure *procedure-name* must have both a valid LANGUAGE clause and a valid EXTERNAL clause.

Severity: 8 (error)

System action: The statement cannot be executed.

User response: Correct and reissue the failing statement.

DSNH4001I HOST LANGUAGE STATEMENT EXCEEDS MAXIMUM OF 1000 LINES

Explanation: A PL/I program contains a declaration that covers more than 1000 source lines and contains a LOB variable declaration.

Severity: 8

System action: Processing continues.

User response: Split the DCL statement into two parts.

DSNH4408I E *csectname* **LINE** *nnnn* **COL** *cc* **THE VALUE IS NOT COMPATIBLE WITH THE DATA TYPE OF ITS TARGET**

Explanation: The data type of the value to be assigned to the parameter or SQL variable is incompatible with the declared data type of the assignment target. Both must be:
- Numerals
- Characters
- Graphics
- Dates or character
- Times or character
- Timestamps or character
- Row ID's
- The same distinct types

Severity: 8 (error)

System action: The statement cannot be processed.

User response: Examine the current definition for the parameter or SQL variable and ensure that it has the proper data type.

DSNH4700I E *csectname* **LINE** *nnnn* **COL** *cc* **ATTEMPT TO USE NEW FUNCTION WITH NEWFUN NO**

Explanation: When the SQL processing option NEWFUN has a value of NO, you cannot use syntax for functions that this release of DB2 introduces. An attempt was made to use one of these functions.

Severity: 8 (error)

System action: If the containing SQL statement is a declaration, it is discarded. If the containing SQL statement is executable, it is replaced by host language statements that will, if executed, set an error code in the SQLCA.

Programmer response: Do one of these things:
- Restrict your program to functions that earlier releases of DB2 introduced. These functions are allowed regardless of the value of NEWFUN.
- Process your program with a value of YES for the NEWFUN option. However, you cannot BIND the DBRM until New Function Mode has been enabled.

DSNH4701I S *csectname* **OPTION:** *option* **OPTION NOT PERMITTED WITH NEWFUN NO**

Explanation: You cannot use the indicated SQL processing option when the SQL processing option NEWFUN has a value of NO.

Severity: 12 (severe error)

System action: The precompiler is terminated.

Programmer response: To prevent this error in the future, do one of the following:
- Restrict your precompilation to options that were introduced in earlier releases of DB2. These options are valid regardless of the value of NEWFUN.
- Process your program with a value of YES for the NEWFUN option.

DSNH4702I *column* **IS NOT VALID IN THE CONTEXT WHERE IT IS USED**

Explanation: The specified column is not a column of the table or view that was specified as the object of the UPDATE or INSERT statement.

System action: The statement cannot be executed. No data was updated.

Programmer response: Verify that the column and table names are specified correctly in the SQL statement. If a correlation name is specified for a table or view, verify that subsequent table references of the table or view in qualified column names use the correlation name and not the table or view name.

SQLSTATE: 42703

DSNH4760I THE DB2 SQL COPROCESSOR IS USING THE LEVEL *nnn* **INTERFACE UNDER DB2 VERSION** *DB2-version*

Explanation: This message specifies the level of the DB2 SQL coprocessor that is in use. This message is for IBM software support.

Severity: 0 (informational)

System action: No action required.

User response: No action required.

DSNH4761I THE DB2 SQL COPROCESSOR HAS CHANGED THE NEWFUN OPTION VALUE FROM YES TO NO

Explanation: The DB2 SQL coprocessor option NEWFUN(YES) was specified, but the level of the DB2 SQL coprocessor interface is not compatible with NEWFUN(YES). Therefore, the DB2 SQL coprocessor is running with NEWFUN(NO). Functions that are specific to the new version of DB2 are not allowed.

Severity: 4 (warning)

System action: No action required.

User response: Upgrade to a later compiler version that supports the new DB2 functions.

DSNH4764I I *csect-name* **SQL OPTION** *option* **IS IN EFFECT**

Explanation: The DB2 coprocessor will run with SQL option *option*

Severity: 0 (informational)

System action: Processing continues.

User response: If *option* is CCSID(*nnnn*): If the value *nnnn* is not the expected CCSID value, explicitly specify the correct CCSID and recompile the program.

DSNH4771I E *csectname* **PRECOMPILER OPTION QUOTE CANNOT BE SPECIFIED IN THIS CONTEXT. USE THE APOST OPTION**

Explanation: When the value of the CCSID precompiler option is 1026 or 1155, and the value of the HOST precompiler option is IBMCOB, the APOST precompiler option must be used because the precompiler modified source defines the quotation mark with the hex point of 'FC'X, which is not taken by the COBOL compiler.

Severity: 8 (error)

System action: Processing continues.

User response: Specify the APOST precompiler options, and precompile the application again. There is no need to modify the COBOL source.

DSNH4777I E *csectname* **LINE** *nnnn* **COL** *cc* **NESTED COMPOUND STATEMENTS NOT ALLOWED**

Explanation: An SQL procedure contains nested compound statements, which are not allowed.

Severity: 8 (error)

System action: The statement cannot be executed.

User response: Rewrite the SQL procedure body so that it does not contain nested compound statements.

DSNH4778I E *csectname* **LINE** *nnnn* **COL** *cc* **END LABEL** *label-name* **NOT SAME AS BEGIN LABEL**

Explanation: An SQL procedure statement cantains an ending label and a beginning label that do not match.

Severity: 8 (error)

System action: The statement cannot be executed.

User response: Change the ending label in the statement to match the beginning label.

DSNH4779I *csectname* **LINE** *nnnn* **COL** *cc* **LABEL** *label* **SPECIFIED** *statement* **IS NOT VALID**

Explanation: In an SQL procedure, *label* is specified on a GOTO or LEAVE statement. *label* is not defined or is not a valid label for the statement.

The label on a LEAVE statement must be the label for a LOOP, REPEAT, WHILE, or compound statement.

The label in a GOTO statement must be defined within a scope that the GOTO statement can reach.

- If the GOTO statement is defined in a compound statement, *label* must be defined inside the same compound statement.
- If the GOTO statement is defined in a handler, *label* must be defined in the same handler and must follow the other scope rules.
- If the GOTO statement is defined outside a handler, *label* must not be defined within a handler.

Severity: 8 (error)

System action: The statement cannot be processed.

User response: Change the label in the LEAVE statement to match the label on the LOOP, REPEAT, WHILE, or COMPOUND statement that contains the LEAVE statement.

Ensure that the label in the GOTO statement is defined within a scope that the GOTO statement can reach.

DSNH4780I **E** *csectname* **LINE** *nnnn* **COL** *cc* **UNDO SPECIFIED FOR A HANDLER AND ATOMIC NOT SPECIFIED**

Explanation: In an SQL procedure, a compound statement is defined as NOT ATOMIC, but the compound statement contains an UNDO handler. An UNDO handler can be used only for a compound statement that is defined as ATOMIC.

Severity: 8 (error)

System action: The statement cannot be executed.

User response: Define the compound statement as ATOMIC, or change the UNDO handler to a CONTINUE or EXIT handler.

DSNH4781I **E** *csectname* **LINE** *nnnn* **COL** *cc* **CONDITION** *condition-name* **SPECIFIED IN HANDLER NOT DEFINED**

Explanation: In an SQL procedure, a handler is declared for condition *condition-name*, but the SQL procedure does not contain a condition declaration statement that defines *condition-name*.

Severity: 8 (error)

System action: The statement cannot be executed.

User response: Include a condition declaration statement in the SQL procedure that relates *condition-name* to an SQLSTATE value.

DSNH4782I **E** *csectname* **LINE** *nnnn* **COL** *cc* **CONDITION VALUE** *value* **SPECIFIED IN HANDLER NOT VALID**

Explanation: In an SQL procedure, a condition handler is not valid for one of the following reasons:

- The handler specifies an SQLSTATE value that is not valid.
- The handler specifies duplicate conditions.
- The handler specifies SQLWARNING, SQLEXCEPTION, or NOT FOUND with other conditions.

Severity: 8 (error)

System action: The statement cannot be executed.

User response: Specify a valid condition in the handler. Ensure that a handler specifies a condition only once.

DSNH4783I **E** *csectname* **LINE** *nnnn* **COL** *cc* **SELECT LIST FOR CURSOR** *cursor-name* **IN FOR STATEMENT NOT VALID**

Explanation: An SQL procedure contains a FOR statement in which the select list in the cursor declaration has a column that is not valid. That column is a duplicate of another column in the select list, or the column is not named.

Severity: 8 (error)

System action: The statement cannot be executed.

User response: Ensure that all columns in the select list for a cursor declaration in a FOR statement are unique and are named. If a column of the select list is the result of a column function, specify an AS clause to name the column.

DSNH4784I **E** *csectname* **DSNHDECP IS FOR DB2 VERSION** *"dsnhdecp-version"* **BUT** *product-name* **IS FOR DB2 VERSION** *"dsnhdecp-version"*

Explanation: The release information from the given DSNHDECP does not match the current level of the DB2 precompiler or SQL coprocessor.

Severity: 12 (severe error)

System action: The precompiler or SQL coprocessor is terminated.

Programmer response: Ensure that a DSNHDECP data-only load module for the correct DB2 version is available to the precompiler or SQL coprocessor, and that DSNHDECP load module is the one that DB2 finds first. DB2 searches for the DSNHDECP module in the following places, and in the following order:

- The STEPLIB concatenation for the precompile or compiler step for SQL coprocessor
- The JOBLIB concatenation for the job that includes the precompile step or compiler step for SQL coprocessor
- The system link list

DSNH4785I E *csectname* **LINE** *nnnn* **COL** *cc* **THE DECLARATION OR USE OF THE SQLSTATE OR SQLCODE VARIABLE IS NOT VALID**

Explanation: The declaration or use of an SQLCODE or SQLSTATE special variable is not valid for one of the following reasons:

- SQLSTATE is declared with a data type other than CHAR(5).
- SQLCODE is declared with a data type other than INTEGER.
- NULL is assigned to SQLCODE or SQLSTATE.

Severity: 8 (error)

System action: The statement cannot be processed.

User response: Declare SQLSTATE only as CHAR(5) and SQLCODE only as INTEGER. Do not assign the null value to SQLCODE or SQLSTATE.

DSNH4787I E *csectname* **LINE** *nnnn* **COL** *cc* **RESIGNAL STATEMENT NOT WITHIN A HANDLER**

Explanation: The RESIGNAL statement can only be used inside condition handlers.

Severity: 8 (error)

System action: The statement cannot be processed.

User response: Remove the RESIGNAL statement or use a SIGNAL statement instead.

DSNH4788I *csectname* **CCSID** *ccsid* **FOR PROGRAM SOURCE CODE IS NOT SUPPORTED**

Explanation: The DB2 precompiler or DB2 coprocessor does not accept input source code with the specified CCSID. The CCSID must be an EBCDIC CCSID.

Severity: 16 (unrecoverable error)

System action: The precompiler or DB2 coprocessor terminates.

User response: Specify an EBCDIC CCSID.

DSNH4911I E *csectname* **SQL TYPE OF HOST VARIABLE IS INVALID**

Explanation: SQLGALHV received a host variable with an invalid SQL data type code.

Severity: 8 (error)

System action: The invalid host variable is ignored.

User response: Correct the SQL data type code and register the host variable again.

DSNH4912I E *csectname* **LENGTH OF HOST VARIABLE IS OUT OF RANGE**

Explanation: SQLGALHV received a host variable with an invalid data area. The length of the data area in parameter SQL_LENGTH is either too large or too negative.

Severity: 8 (error)

System action: The host variable is ignored.

User response: Correct the data area length parameter and register the host variable again.

DSNH4913I E *csectname* **TOKEN IDENTIFIER HAS ALREADY BEEN USED**

Explanation: The token identifier passed to SQLGALHV already exists for another host variable.

Severity: 8 (error)

System action: The host variable is ignored.

User response: Assign a different token identifier and register the host variable again.

DSNH4914I E *csectname* **INVALID TOKEN IDENTIFIER**

Explanation: The token identifier passed to SQLGALHV is invalid.

Severity: 8 (error)

System action: The host variable is ignored.

User response: Assign a token identifier that is a valid 4-byte integer, and register the host variable again.

DSNH4915I E *csectname* **SQLGINIT HAS ALREADY BEEN INVOKED**

Explanation: A call to SQLGINIT has already been made.

Severity: 8 (error)

System action: Control is returned to caller with no other action taken.

User response: Change the compiler or precompiler code so that SQLGINIT is only called once. The compilation or precompilation currently in progress may continue.

DSNH4916I E *csectname* **SQLGINIT HAS NOT BEEN INVOKED**

Explanation: A call to SQLGINIT has not been made.

Severity: 8 (error)

System action: Control is returned to caller with no other action taken.

User response: Change the compiler or precompiler code so that SQLGINIT is called before any of the other API entry points.

DSNH4919I E *csectname* **TASK ARRAY TOO SMALL**

Explanation: The DB2 coprocessor needs more task array entries than the coprocessor allocated.

Severity: 8 (error)

System action: Control is returned to caller with no other action taken.

User response: Allocate a larger task array and make the call to SQLGCMPL again.

DSNH4920I E *csectname* **TOKEN ID ARRAY TOO SMALL**

Explanation: The DB2 coprocessor needs more token identifier array entries than the coprocessor allocated.

Severity: 8 (error)

System action: Control is returned to caller with no other action taken.

User response: Allocate a larger token identifier array and make the call to SQLGCMPL again.

DSNH4941I U *csectname* **BLANK OR EMPTY SQL STATEMENT TEXT**

Explanation: The SQL statement passed to SQLGCMPL was blank.

Severity: 8 (error)

System action: Control is returned to caller with no other action taken.

User response: Assign the SQL statement buffer to the correct parameter and call SQLGCMPL again.

DSNH4999I U *csectname* **PRECOMPILER SERVICES INTERNAL ERROR**

Explanation: The DB2 coprocessor encountered an unrecoverable internal error.

Severity: 16 (unrecoverable error)

System action: Control is returned to caller with no other action taken.

User response: Terminate compilation.

DSNH5011I HOST VARIABLE ARRAY *host-variable-array* **IS EITHER NOT DEFINED OR IS NOT USABLE**

Explanation: Host-variable-array *host-variable-array* was specified in a multiple-row FETCH or multiple-row INSERT statement. The host-variable-array is not defined or is not usable for one of the following reasons:

- The host variable is not a valid host-variable-array.
- The host variable is not a dimensioned array.
- The host-variable-array has more than 1 dimension.
- The host-variable-array defines a structure that does not conform to the rules for defining a host-variable-array.
- The dimension of the host-variable-array used for indicators is not equal to the dimension of the main host-variable-array.

System action: The statement cannot be processed.

Programmer response: Correct any of the following and precompile the program again. Ensure that:

- the dimension of the indicator variable array is equal to the dimension of the main host-variable-array.
- the host-variable-array is a single-dimensional array.
- the host-variable-array or host indicator variable array do not contain structures that are not permissible. The only structures that may be defined in a host-variable-array are those that are used to define variable length string host variables.

See the *DB2 Application Programming and SQL Guide* for more language-specific information on how to define the host-variable-arrays in a program.

SQLSTATE: \

DSNH5033I DECLARE VARIABLE NOT VALID FOR *variable*

Explanation: Host variable *variable* specified in a DECLARE VARIABLE statement is not valid. The DECLARE VARIABLE statement occurs after any SQL statement (in a COBOL or FORTRAN application program) or after a DECLARE CURSOR statement (in an application program in any other host language), where the SQL statement and the DECLARE VARIABLE refer to the same variable.

Severity: 8 (error)

System action: The statement cannot be processed.

User response: Ensure that the DECLARE VARIABLE occurs before the use of the host variable in an SQL statement in the application.

DSNH5034I DECLARE VARIABLE NOT VALID FOR *variable*

Explanation: Host variable *variable* specified in a DECLARE VARIABLE statement is not valid for one of the following reasons:

- A reference is made to a variable that does not exist.
- A reference is made to a variable that is a structure.
- A reference is made to a variable that is a component of two more structures.
- The host variable is not defined as character, varying-length character, graphic, varying-length graphic, CLOB, or DBCLOB. In this case, CLOB AS LOCATOR or DBCLOB AS LOCATOR is not considered a CLOB or DBCLOB.
- FOR BIT DATA, CCSID 65535, FOR MIXED DATA, FOR SBCS DATA is specified and the variable is defined as graphic, varying-length graphic, or DBCLOB.
- FOR BIT DATA, or CCSID 65535 is specified and the variable is defined as CLOB.

Severity: 8 (error)

System action: The statement cannot be precompiled.

User response: Correct the definition of the host variable. Precompile the program again.

DSNH5035I HOST VARIABLE *variable* **IS SPECIFIED TWICE IN A DECLARE VARIABLE STATEMENT**

Explanation: The host variable *variable* was specified in two or more DECLARE VARIABLE statements, or was specified two or more times in one DECLARE VARIABLE statement. A host variable can be specified once at most in a DECLARE VARIABLE statement.

Severity: 8 (error)

System action: The statement cannot be processed.

User response: Specify the host variable at most once in a DECLARE VARIABLE statement.

DSNH5036I CCSID VALUE OUT OF RANGE

Explanation: The CCSID for a DECLARE VARIABLE or CAST statement is outside the range of 1 to 65533, or the the value 65535. The value must be within the range of 1 to 65533, or the value must be 65535.

Severity: 8 (error)

System action: The statement cannot be processed.

User response: If this message is issued for a "DECLARE VARIABLE CCSID integer"or "CAST CCSID integer" statement, change the value to be within the range of 1 to 65533 or the value 65535. If this message is issued for a DECLARE VARIABLE statement with the "FOR SBCS DATA" or "FOR MIXED DATA" clause, then the CCSIDs specified on panel DSNTIPF at installation time need to be updated with valid values.

DSNH5037I DECLARE VARIABLE STATEMENT FOUND WITH ONE-PASS PRECOMPILATION

Explanation: A DECLARE VARIABLE statement was found, but the source program is being precompiled with only one pass. If a DECLARE VARIABLE is used in an assembler source program, the ONEPASS precompiler option must not be used. If a DECLARE VARIABLE is used in a C, C++, or PL/I source program, the TWOPASS precompiler option must be used.

Severity: 8 (error)

System action: The source program cannot be precompiled.

User response: Specify the TWOPASS precompiler option or, for assembler, do not specify the ONEPASS precompiler option.

DSNH20008I UNSUPPORTED OPTION *keyword* **SPECIFIED**

Explanation: *keyword* is a deprecated feature that was supported in releases prior to DB2 Version 8, and is no longer supported.

For indexes, only one type is currently supported — type 2.

Severity: 8 (error)

System action: The statement cannot be executed.

User response: Correct the syntax of the SQL statement to remove reference to the unsupported keyword. Refer to the *DB2 SQL Reference* for more information.

SQLSTATE: 560A9

DSNH20060I E *csectname* **LINE** *nnnn* **COL** *cc* **UNSUPPORTED DATA TYPE** *data-type* **ENCOUNTERED IN SQL** *object-type object-name*

Explanation: *data-type* was specified in the definition of *object-name*. *object-type* is an SQL procedure parameter or variable. *data-type* is not supported for SQL procedure parameters or variables.

You can use the same built-in data types for SQL procedure parameters or variables that you can use for the CREATE TABLE statement, except these:

- LONG VARCHAR
- LONG VARGRAPHIC
- ROWID
- distinct type

Severity: 8 (error)

System action: The statement cannot be executed.

User response: Change the syntax to specify one of
the supported data types. Instead of a LONG
VARCHAR data type, use a VARCHAR data type with
an explicit length. Instead of a LONG VARGRAPHIC
data type, use a VARGRAPHIC data type with an
explicit length.

DSNH20061I E *csectname* **LINE** *nnnn* **COL** *cc* **UNEXPECTED ERROR RETURNED FROM LANGUAGE ENVIRONMENT: REASON CODE** *reason-code*, **RETURN CODE** *return-code module-name*

Explanation: An Language Environment error occurred while the DB2 precompiler was processing an SQL procedure. The reason codes and associated return codes are:

Reason code	Meaning and associated return code
1	No PIPI token. *return-code* is the return code from the CEEPIPI(init_sub) call.
2	CEE could not be loaded. *return-code* is the return code from the CEEPIPI(add_entry) call.
3	PIPI would not terminate. *return-code* is the return code from the CEEPIPI(term) call.
4	Call to *module-name* failed. *return-code* is the return code from the CEEPIPI(call_sub) call.
5	Bad response from PIPI. *return-code* is the return code from *module-name*.

See the explanation of return codes for the appropriate CEEPIPI call in *z/OS Language Environment Programming Guide* for explanations of the Language Environment return codes.

Severity: 8 (error)

System action: The statement cannot be executed.

User response: Correct the condition that is described by *reason-code*.

DSNH20129I E *csectname* **LINE** *nnnn* **COL** *cc* **LOCAL SPECIAL REGISTER IS NOT VALID AS USED**

Explanation: Local special registers cannot be referenced in an assignment statement (SET or VALUES INTO) that assigns multiple values. For example, the following special registers are local: CURRENT SERVER and CURRENT PACKAGESET.

Additionally, the local special registers cannot be referenced in a SET host-variable statement that has parenthesis around the special register.

The following statements are not valid:

```
SET (:hv1) = (CURRENT SERVER);
SET (:hv1,:hv2) = (CURRENT SERVER,CURRENT
PATH);
VALUES CURRENT SERVER, CURRENT DATE
INTO :c1, :c2;
```

Severity: 8 (error)

System action: The statement cannot be executed.

User response: Split the statement into multiple statements so that the local special registers are not referenced in a statement that assigns multiple values. Also be sure that a SET statement for a local special register does not use parenthesis on either side of the equal sign.

| **DSNH20289I INVALID STRING LENGTH UNIT** *unit* **IN EFFECT FOR FUNCTION** *function-name*

| **Explanation:** The statement invoked a built-in function using a string length unit that is not valid with the type of data provided for the function. This error can occur for the following situations:

- Non-string data was specified with an explicit length unit *unit* for the LENGTH function. A string length unit cannot be specified for non-string data.
- Bit data or binary string data was specified with *unit*. CODEUNITS16, or CODEUNITS32 are not valid with bit data or binary string data.
- Graphic data was specified with OCTETS as the string length unit. OCTETS is not valid with graphic data.

| **System action:** The statement cannot be processed.

| **System programmer response:** Change the invocation of the function to remove the invalid string length unit specification or change it to a valid string length unit for the data type being processed.

DSNH20367I W *csectname* **LINE** *nnnn* **COL** *cc* **OPTION** *clause* **IS NOT SUPPORTED IN THE CONTEXT IN WHICH IT WAS SPECIFIED**

Explanation: The *clause* in the statement is not
supported in the context in which has been specified.
The *clause* is ignored.

System action: Processing continues normally.

User response: No change is required for the current
release; however, it is recommended that the statement
be modified to conform to valid SQL syntax.

Chapter 8. DSNI... messages

DSNI001I RESTART HAS BEEN DEFERRED REASON reason TYPE type NAME name

Explanation: Restart could not be performed for one or more pages of the resource identified by *type* and *name*. Those pages are placed in the logical page list (LPL) and restart for those pages is deferred. The reason code explains why restart was not performed.

Refer to Table 2 on page 662 for an explanation of resource type codes.

System action: DB2 restart continues. Pages for which restart activity was not performed are put into the LPL.

User response: Correct the problem indicated by the reason code. Then use the START DATABASE command or the RECOVER utility to remove the pages from the LPL.

DSNI002I *csect-name* **UNABLE TO START OBJECT TYPE** *object-type* **OBJECT NAME** *object-name* **REASON** *reason* **TYPE** *type* **NAME** *name*

Explanation: The resource identified could not be started because it was unavailable. The resource could be a database, a table space, an index space, or a partition within a table space or an index space. The 'object-type' identifies the type. The 'object-name' identifies the name. If the DB2 catalog is unavailable, 'object-type' contains binary zeros and 'object-name' contains blanks.

The resource that was unavailable is identified by 'type' and 'name'. The 'type' code identifies the type of the resource as well as the format and content of the 'name'. For an explanation of resource type codes, see “Resource types” on page 662. The 'reason' contains the reason code explaining why the resource was unavailable.

System action: The specified database, space, or partition is not started and remains in its current state. If there are other databases, spaces, or partitions to be started, they are processed normally.

User response: Correct the resource unavailable problem and reissue the -START command.

DSNI003I *csect-name* **UNABLE TO STOP OBJECT TYPE** *object-type* **OBJECT NAME** *object-name* **REASON** *reason* **TYPE** *type* **NAME** *name*

Explanation: The identified object could not be stopped because a resource required to perform the stop was unavailable. The object could be a database, a table space, an index space, or a partition within a table space or an index space.

The 'object-type' identifies the type and 'object-name' identifies the name. If the DB2 catalog is unavailable, 'object-type' contains binary zeros and 'object-name' contains blanks.

The resource that was unavailable is identified by 'type' and 'name'. The 'type' code identifies the type of the resource as well as the format and content of the resource, 'name'. For an explanation of resource type codes, see “Resource types” on page 662. The 'reason' contains the reason code explaining why the resource was unavailable.

System action: The specified database, space, or partition is not stopped and remains in the current state. If there are other databases, spaces, or partitions to be stopped, they are processed normally.

User response: Correct the resource unavailable problem and reissue the -STOP command.

DSNI004I RESOURCE HAS BEEN STOPPED TYPE type NAME name

Explanation: The resource identified by TYPE and NAME could not be opened/closed, initialized, or was found to be in an inconsistent state. The resource has been 'stopped' by DB2. If an open error occurred during DB2 startup, restart processing for the resource was deferred.

TYPE and NAME identify the resource that has been stopped. The TYPE code identifies the type of the resource as well as the format and content of the resource, NAME.

For an explanation of resource type codes, see “Resource types” on page 662.

System action: The database, table space or index space is stopped by DB2.

User response: Previous messages provide information to allow you to determine the cause of the problem and the method of correcting it. Refer to them, and correct the problem. Then restart the database, table space, or index space using the -START command; and invoke the application again.

DSNI005I *csect-name* **LPL OR GRECP RECOVERY CANNOT BE DONE FOR** *function* **- ID =** *id* **OBJECT TYPE** *object-type* **OBJECT NAME** *object-name* **REASON** *reason* **TYPE** *type* **NAME** *name*

© Copyright IBM Corp. 1983, 2008

Explanation: The resource that is identified could not be recovered during a recovery function that is identified by *function_id* because of problems during LPL or GRECP recovery. The resource could be a table space, an index space, or a partition within a table space or index space. *object-type* identifies the type of resource. *object-name* identifies the name of the resource.

The resource that could not be recovered is identified by *type* and *name*. The *type* code identifies the type of the resource, as well as the format and content of *name*. *reason* contains the reason code explaining why the resource could not be recovered. For an explanation of resource type codes, see "Resource types" on page 662. *reason* contains the reason code explaining why the resource could not be recovered.

The potential values for *function* are START DATABASE COMMAND, AUTOMATIC GRECP RECOVERY, or AUTOMATIC LPL RECOVERY.

id is a decimal number that represents the recovery function that is indicated by *function*.

This message, in combination with message DSNI006I or DSNI022I, provides the LPL or GRECP recovery information for a recovery function that is identified by the *recovery-id*.

System action: The specified space, or partition is started but remains in a LPL/GRECP state. If there are other spaces or partitions to be recovered, they are processed normally.

User response: Correct the problem that is specified by the reason code, and reissue the -START command.

DSNI006I *csect-name* **ASYNCHRONOUS GRECP OR LPL RECOVERY IN PROGRESS FOR** *function* **- ID** *id* **TYPE** *type* **NAME** *name*

Explanation: This message is issued in response to a START DATABASE command, AUTOMATIC GRECP RECOVERY, or AUTOMATIC LPL RECOVERY function under the following conditions:

- When the object (table space or index space), which is identified by *type* and *name*, had group buffer pool recovery pending (GRECP) or logical page list (LPL) status, and
- Recovery was triggered.

The values for *function* could be START DATABASE COMMAND, AUTOMATIC GRECP RECOVERY, or AUTOMATIC LPL RECOVERY.

id is a decimal number that represents the recovery function.

For an explanation of resource type codes, refer to "Resource types" on page 662 for an explanation of resource type codes.

System action: The object is started. Also, GRECP or LPL recovery is initiated. During LPL or GRECP recovery process, message DSNI022I might be issued periodically to indicate the recovery status. If the recovery completes successfully, message DSNI021I will appear at the end of recovery. Otherwise, message DSNI005I will indicate the error.

User response: Wait for the DSNI021I or the DSNI005I message to be issued to the console. You can periodically issue a DISPLAY DATABASE RESTRICT command for the specified table space or index space to determine whether the GRECP or LPL exception state has been cleared.

DSNI007I *csect-name* **UNABLE TO START/RECOVER BECAUSE UNABLE TO DIRECT READ A LOG RECORD LOGRBA** *logrba*

Explanation: Unable to START (or RECOVER) because the necessary log records could not be accessed. Either an error occurred while attempting to allocate or open the log data set or an expected log record was not found on the log.

If the message is in response to a START command, issue the -DISPLAY command to determine which resource(s) was not started. If the message is in response to a RECOVER utility, the resource is the table space or index space that is the object of the RECOVER.

LOGRBA gives the log RBA of the log record that is unavailable.

System action: The resource is not started (or recovered) and remains in the current state. If this is a START command, the resources are processed normally if there are other databases or spaces to be started and they do not involve reading the log.

User response: Resubmit the utility, or reissue the START command, when the problem with the log has been resolved.

System programmer response: Run the print log map (DSNJU004) utility. This may be run with DB2 up or down. Find the data set that contains the requested RBA in the output of DSNJU004. If the required RBA is not in a listed data set, one of the following may have occurred:

- The data set was deleted from the bootstrap data set by a run of the change log inventory (DSNJU003) utility. In this case, rerun the utility to insert the missing data set name. DB2 must be down to run this utility.
- The bootstrap data set has been damaged. In this case, the BSDS must be restored from the dump that occurred with the most recent archive log data set. DSNJU003 must be run to supply changes that

occurred during and after the last off-load operation. Changes may be determined from the operator's console listing.

If a listed log data set contains the required RBA, this message will be preceded by one or more error messages from the log manager (DSNJ). See these messages for appropriate action.

Problem determination: Issue the -DISPLAY command to verify what got started.

DSNI008I DATABASE VALIDATION PROCEDURE name COULD NOT BE LOADED

Explanation: The database validation procedure identified by 'name' could not be loaded.

User response: Correct the name, and rerun the application.

DSNI009I *csect-name* **DATABASE EDIT PROCEDURE** *name* **COULD NOT BE LOADED**

Explanation: The database edit procedure identified by 'name' could not be loaded.

User response: Correct the name, and rerun the application.

DSNI010I BROKEN PAGE ACCESSED TYPE type NAME name MODNAME csect-name CONN-ID id CORR-ID id LUW-ID id

Explanation: The page identified by TYPE and NAME could not be accessed, because it is marked as logically broken.

The TYPE and NAME identify the resource that is currently unavailable. The TYPE code identifies the type of the resource as well as the format and content of the resource NAME.

Type codes '00000301', '00000302', and '00000303' apply to this message. Refer to Appendix A, "Problem determination," on page 659 for a table showing all types and names.

'csect-name' has encountered the logically broken page. CONN-ID, CORR-ID, and LUW-ID identify the thread involved in the problem. If the LUW-ID is an '*', the thread originated at this site. Correlation IDs beginning with '0nn', where 'nn' is a number from 01 to 28, identify system agents.

User response: If the TYPE is '00000300', use NAME to determine if the page is in an index space or a table space. Use the REPAIR utility to fix the broken page. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNI011I PAGE TEMPORARILY UNAVAILABLE TYPE type NAME name MODNAME modname ERQUAL erqual

Explanation: The page identified by TYPE and NAME has been temporarily marked as logically broken to prevent access until it can be recovered.

The TYPE and NAME identify the resource that is currently unavailable. The TYPE code identifies the type of the resource as well as the format and content of the resource NAME. Refer to Table 2 on page 662 for an explanation of resource type codes.

The module named modname has called a page checking routine, which discovered an error in the page.

The ERQUAL gives the trace id for the error in the page. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

System action: Automatic recovery for the page has been scheduled. The current agent will abend with reason code '00C90105', and a dump will be taken. The page will be unavailable to this and other agents until the online recovery has completed successfully. Message DSNI010I will be issued, and a 'resource unavailable' condition will be returned to any agents that attempt to access the page. If the online recovery fails, the page will be marked as broken until it is repaired or recovered (see message DSNI012I).

User response: Save the dump. Watch for message DSNB215I (indicating successful recovery of the page), or DSNI012I (indicating failure of recovery of the page).

DSNI012I PAGE LOGICALLY BROKEN TYPE type NAME name MODNAME modname ERQUAL erqual

Explanation: The page identified by TYPE and NAME is logically incorrect. Additional attempts to access the page will be denied.

The TYPE and NAME identify the resource that is currently unavailable. The TYPE code identifies the type of the resource as well as the format and content of the resource NAME. Refer to Table 2 on page 662 for an explanation of resource type codes.

The module named 'modname' has called a page checking routine, which discovered an error in the page.

The ERQUAL gives the trace id for the error in the page. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem. An ERQUAL of X'0000' indicates that an abend occurred during online recovery or recover utility. The page has been marked as broken, and the recovery has failed.

System action: The page is marked as logically

broken, an 04E abend with reason code '00C90102' is issued, and a dump is taken. The page will be unavailable to this and other agents until it is repaired.

User response: Save the dump for analysis by IBM Support Center. Copy the table space or index space to another data set for later analysis by IBM Support Center. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Operator response: Notify the system programmer.

System programmer response: See the Explanation section of this message.

DSNI013I *csect-name* **POTENTIALLY INCONSISTENT DATA REASON** *reason* **ERQUAL** *erqual* **TYPE** *type* **NAME** *name* **CONN-ID** *id* **CORR-ID** *id* **LUW-ID** *id*

Explanation: This message identifies a page with potentially inconsistent data. This message is issued for each page involved in the current data manager operation. The 'csect-name' in the message identifies the CSECT that issued the message.

The 'erqual' in the message identifies the specific abend in the CSECT that caused this message to be issued. This information and the reason code are used by IBM Support Center to identify the cause of the problem.

'name' and 'type' identify the page and its type. The TYPE code identifies the type, as well as the format and content, of the NAME. Refer to Appendix A, “Problem determination,” on page 659 for a table showing all types and names.

CONN-ID, CORR-ID, and LUW-ID together identify the thread involved in the problem. If the LUW-ID is an '*', the thread originated at this site. Correlation IDs beginning with '0nn' where 'nn' is a number from 01 to 28, identify system agents.

For further information, see the Explanation section for the abend reason code specified in this message.

System action: The job is abended shortly after this message appears.

User response: Notify the system programmer.

Operator response: Collect the console output from the system on which the job was run. Save this output for use in problem determination.

System programmer response: See the Problem Determination section for the specific abend reason code.

Problem determination: See the System Programmer Response section of the abend reason code specified in this message.

DSNI014I *csect-name* **DATA IN USE DURING ABEND REASON 00C90101 ERQUAL** *erqual* **TYPE** *type* **NAME** *name* **CONN-ID** *id* **CORR-ID** *id* **LUW-ID** *id*

Explanation: This message identifies a page that was in use at the time an abend condition occurred with reason code 00C90101. This message is issued for each page involved in the current operation.

ERQUAL identifies the specific abend in the CSECT that caused this message to be issued. This information and the reason code are used by IBM Support Center to identify the cause of the problem.

NAME and TYPE identify the page and its type. The TYPE code identifies the type, as well as the format and content, of the NAME. Refer to Table 2 in Appendix A, “Problem determination,” on page 659 for an explanation of resource type codes.

CONN-ID, CORR-ID, and LUW-ID together identify the agent involved in the problem. If the LUW-ID is an '*', the thread originated at this site. Correlation IDs beginning with '0nn' where 'nn' is a number from 01 to 28, identify system agents.

System action: The job abends with reason code 00C90101 after this message appears.

User response: Notify the system programmer.

Operator response: Collect the console output from the system on which the job was run for the period of time spanning the failure. Save this output for use in problem determination.

System programmer response: See abend reason code 00C90101 for more information.

Problem determination: In general, the data in these pages is fine. However, if problems are reported that repeatedly involve the same pages or if the same pages are reported by other abends, you might want to analyze the set of pages involved.

If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNI015I **OBJECT name IS ALREADY STARTED FOR RW ACCESS. THE OBJECT WILL REMAIN STARTED FOR RW ACCESS**

Explanation: The -START DATABASE command with the ACCESS(FORCE) keyword was entered, but the object specified was already started for read/write access.

DSNI016I **RESOURCE TYPE type NAME name MUST FIRST BE STOPPED BEFORE IT CAN BE STARTED IN A DIFFERENT ACCESS MODE**

Explanation: When issuing the -START DATABASE

command on the ROSHARE OWNER system, the object to be started must first be stopped before changing access modes. For example, if a table space in a database defined as ROSHARE OWNER is started for RW access, before it can be started for RO access it must be stopped via the -STOP DATABASE command. It can then be started for RO access via the -START DATABASE command.

Severity: 8 (error)

System action: The -START DATABASE command is not allowed to proceed.

User response: Issue a -STOP DATABASE command to stop the object before attempting to start it.

DSNI017I RESOURCE TYPE type NAME name IS INCONSISTENT DUE TO ONE OR MORE PENDING STATES

Explanation: There are two possible reasons for this message:

1. The header page of a table space, index, or partition belonging to a database defined as ROSHARE OWNER could not be marked as consistent during the processing of the STOP DATABASE command because one or more of the following was true:
   - The table space, index, or partition is in recovery pending status.
   - The table space, index, or partition is in check pending status.
   - The table space, index, or partition has pages in the logical page list (LPL).
   - The table space, index, or partition status has a utility in progress (UTUT status on the output from a DISPLAY DATABASE command).
   - The table space, index, or partition has a utility in progress that allows read-only access (UTRO on the output from a a DISPLAY DATABASE command).
   - An I/O error range exists.
2. A retained lock is held on the table space, index, or partition. The retained lock could be signifying a pending status.

If the STOP DATABASE command was issued without specifying the SPACENAM parameter, then one or more of the conditions stated above is true for one or more table spaces, indexes, or partitions in the specified database.

Severity: 4 (warning)

System action: For case 1, if the STOP DATABASE command was specified with the SPACENAM parameter, the table space, index, or partition is stopped. DB2 does not allow access to an inconsistent table space, index, or partition by any ROSHARE READ system.

For case 2, no access to an inconsistent table space, index, or partition is allowed until the retained lock is released by the failed system at completion of its restart.

User response: Issue a DISPLAY DATABASE command to display the status of the object in question. Resolve the pending states and then reissue the STOP DATABASE command to ensure that the consistency indicator is reset.

DSNI018I UNABLE TO GET LOCK FOR RESOURCE TYPE type NAME name. WILL RETRY TO GET LOCK.

Explanation: The resource identified by 'type' and 'name' could not be locked because of lock contention. The lock request is made again for this resource.

The 'type' and 'name' identify the resource trying to get a lock. The 'type' code identifies the type of resource as well as the format and content of the resource 'name'.

Refer to Table 2 in Appendix A, “Problem determination,” on page 659 for an explanation of resource type codes.

Severity: 0 (informational)

System action: The lock request is made again for the specified resource.

DSNI019I THE STOP COMMAND FOR OBJECT name WAS OVERRIDDEN BY A START FORCE. THE STOP COMMAND IS IGNORED.

Explanation: The object identified by 'name' was not stopped because a START FORCE command on this object was subsequently issued and processed.

This message is informational only.

Severity: 0 (informational)

System action: The object is not stopped.

DSNI020I *csect-name* **UNABLE TO DRAIN RESOURCE TYPE** *type* **NAME** *name***. WILL RETRY DRAINING THE RESOURCE.**

Explanation: The resource identified by 'type' and 'name' could not be drained. The 'type' code identifies the type of resource as well as the format and content of the resource 'name'. The drain request is made again for this resource.

Refer to Table 2 in Appendix A, “Problem determination,” on page 659 for an explanation of resource type codes.

System action: The drain request is made again for the specified resource.

DSNI021I *csect-name* **GRECP OR LPL RECOVERY FOR** *function* **- ID =** *id* **TYPE** *type* **NAME** *name* **HAS COMPLETED SUCCESSFULLY.**

Explanation: The group buffer pool recovery pending (GRECP) or logical page list (LPL) exception state of the table space or index space has been cleared. The table space or index space can be identified by *type* and *name*.

The potential values for *function* are START DATABASE COMMAND, AUTOMATIC GRECP RECOVERY, or AUTOMATIC LPL RECOVERY.

id is a decimal number that represents a recovery function.

Refer to "Resource types" on page 662 for an explanation of resource type codes.

This message is always preceded by the DSNI006I message, which indicates the beginning of the LPL or GRECP recovery for the object.

System action: The identified table space or index space is started normally.

DSNI022I *csect-name* **LPL** *or* **GRECP RECOVERY LOG APPLY FOR** *function* **- ID =** *id* **AT LOGPOINT** *xxx* **TO LOGPOINT** *yyy*

Explanation: This message periodically shows the progress of the LPL or GRECP recovery of a START DATABASE command, AUTOMATIC GRECP RECOVERY, or AUTOMATIC LPL RECOVERY.

The potential values for *function* are START DATABASE COMMAND, AUTOMATIC GRECP RECOVERY, or AUTOMATIC LPL RECOVERY.

id is a decimal number that represents a recovery function.

xxx is the LOGPOINT of the log record currently being processed. *yyy* is the LOGPOINT of the last log record that will be applied in this recovery. The value of logpoint is RBA in a non-data sharing environment and LRSN in a data sharing environment.

This message, in combination with message DSNI006I, DSNI005I, or DSNI021I, provides the LPL or GRECP recovery information for the recovery function that is identified by *function*.

System action: The LOGAPPLY phase continues normally.

DSNI023I *csect-name* **PAGE SET** *dbnam.psnam* **PART** *part* **IS** *status* **ON BEHALF OF UR** *urid***. BACKOUT TO** *logpoint* **IS REQUIRED.**

Explanation: The value of *status* is either RESTP (Restart Pending) or AREST (Advisory Restart Pending). This message indicates that the named UR has backout work pending against the named page set or partition. The backout work for this UR against this page set or partition will be complete after backward processing of the log reaches the identified logpoint. The value for *logpoint* will be 'RBA *rba*' for non-datasharing and 'LRSN *lrsn*' for datasharing environments.

For non-partitioned page sets, the partition given in the message is the string *n/a*.

DSNI024I *csect-name* **BACKOUT PROCESSING IS COMPLETE FOR PAGE SET** *dbnam.psnam* **PART** *part***.**

Explanation: This message indicates that backout processing is complete for the named table space, index space, or partition. If the object was in restrictive restart pending state, that state is lifted. If the object was in an advisory restart pending state, locks held on the object on behalf of postponed abort URs by the DB2 member issuing the message are released. The advisory restart pending state is lifted if the DB2 member issuing the message is the only member still contributing to the state.

For non-partitioned page sets, the partition given in the message is the string "n/a".

DSNI025I *csect-name* **THE FAST LOG APPLY PROCESS REQUIRES** *aaaaaaaa* **BYTES OF STORAGE, BUT GOT ONLY** *bbbbbbbb* **BYTES**

Explanation: The Fast Log Apply process could not obtain all the required storage. However, the storage acquired is enough for the Fast Log Apply process to continue.

Severity: 0 (informational)

System action: The Fast Log Apply process continues with the smaller amount of storage for the log buffers.

DSNI026I *csect-name* **THE FAST LOG APPLY PROCESS IS DISABLED BECAUSE ALL OF THE ALLOWABLE STORAGE IS EXHAUSTED**

Explanation: There is no ssnmDBM1 storage available for the Fast Log Apply process to use. There are too many jobs (RECOVER or -START DB command) running in parallel.

Severity: 0 (informational)

System action: The recovery of the requested objects continues without the Fast Log Apply process.

DSNI027I *csect-name* **THE FAST LOG APPLY PROCESS IS DISABLED BECAUSE IT IS UNABLE TO ACQUIRE THE MINIMUM REQUIRED STORAGE**

Explanation: There is not enough ssnmDBM1 storage available or reserved for the Fast Log Apply process to use.

Severity: 0 (informational)

System action: The recovery of the requested objects continues without the Fast Log Apply process.

DSNI028I *csect-name* **THE NUMBER OF QUALIFIED LOG RECORDS READ DURING THE FAST LOG APPLY PROCESS IS** *log-records* **AND THE NUMBER OF FAST LOG APPLY BUFFERS PROCESSED ARE** *fla-buffers*

Explanation: This informational message is issued during the Fast Log Apply process. *log-records* indicates the number of log records that have been applied during the Fast Log Apply process. *fla-buffers* indicates the number of buffers used by the Fast Log Apply process.

Severity: 0 (informational)

DSNI029I *csect-name* **THE FAST LOG APPLY PROCESS IS ACTIVE AND THE AVAILABLE** *ssnm***DBM1 STORAGE IS** *bytes* **BYTES**

Explanation: This informational message is issued during the Fast Log Apply process for member *ssnm*. There is sufficient storage for the Fast Log Apply process to continue, and the remaining amount of available storage is displayed in hexadecimal as *bytes*.

Severity: 0 (informational)

DSNI030I *csect-name* **BACKOUT PROCESSING IS CONTINUING BEYOND THE REQUESTED LIMIT DUE TO CATALOG OR DIRECTORY ACTIVITY OF UR** *urid***.**

Explanation: The number of log records requested to be processed during restart's backward phase have been processed. Under other circumstances further backout processing would be postponed. However, because the indicated UR has catalog or directory updates that are not backed out yet (and because backout against catalog or directory page sets cannot be postponed), backout processing is continuing. It will continue until this UR's backout is complete.

System action: Backout processing continues.

DSNI031I *csect-name* **- LOCK ESCALATION HAS OCCURRED FOR RESOURCE NAME =** *name***, LOCK STATE =** *state***, PLAN NAME : PACKAGE NAME =** *id1***:***id2***, COLLECTION-ID =** *id11***, STATEMENT NUMBER =** *id3***, CORRELATION-ID =** *id4***, CONNECTION-ID =** *id5***, LUW-ID =** *id6***, THREAD-INFO =** *id7***:***id8***:***id9***:***id10*

Explanation: Lock escalation balances concurrence with performance by using page locks while a process accesses relatively few pages, then changing to table space or table locks when the process accesses a number of pages that exceeds a defined threshold (LOCKMAX) for the tablespace.

name The name of the escalated resource. The format of *name* depends on the type of table space. For simple or partitioned table spaces, the format is *database-name.tablespace-name*. For a segmented table space, the format is *creator-name.table-name* (or DBID.OBID of the table).

state The lock state (after escalation) of the escalated resource.

id1 The name of the executing plan associated with this thread.

id2 The name of the executing package (if any) associated with this thread.

id3 The statement number of the executing SQL statement associated with this plan.

id4 The correlation identifier of the thread associated with the escalated resource.

id5 The connection identifier of the thread associated with the escalated resource.

id6 The logical-unit-of-work identifier of the thread associated with the escalated resource.

id7 Thread Info - the primary authorization_id associated with the thread.

id8 Thread Info - Work station name of the end user's work station. An asterisk (*) indicates that it is not relevant to the thread or was not provided by the client.

id9 Thread Info - User-Id of the end user. An asterisk (*) indicates that it is not relevant to the thread or was not provided by the client.

id10 Thread Info - Application name of the end user's application. An asterisk (*) indicates that it is not relevant to the thread or was not provided by the client.

id11 The collection-id of the package (if any). resource.

System action: Escalation has occurred for the specified object

DSNI032I *csect-name* **CANCEL THREAD NOBACKOUT COMMAND FAILED FOR THE THREAD =** *token* **REASON =** *reason*

Explanation: DB2 displays this message when it cannot grant a request to cancel a thread without backing out data changes. The reason code explains why the request was rejected.

token Identifies a thread whose processing you requested to cancel. The token is a 1- to 15-digit decimal number.

reason Indicates the reason why the command failed.

System action: Processing terminates normally.

User response: The reason code specifies the action to be taken.

| **DSNI033I** *csect-name* **PAGE SET** *dbname.psname* **PART** *part* **IS MARKED REFP AND** *status* **ON BEHALF OF UR** *ur-id***. RECOVERY TO** *logout* **IS REQUIRED.**

Explanation: No further backout processing will be attempted on the specified page set or partition because backout for unit of recovery *ur-id* has been cancelled. The specified page set or partition is marked Refresh pending (REFP).

Severity: 0 (informational)

User response: If the object in Refresh pending status is a table space, you can resolve the REFP,LPL state of the specified page set or partition in more than one way. You can run the RECOVER utility to perform point-in-time recovery, or run the LOAD utility with the REPLACE option to resolve the REFP,LPL state.

If page set is an index space, run the REBUILD utility to perform point-in-time recovery and resolve the REFP,RBDP or REFP,LPL status. You can use the RBA;LRSN value indicated in this message for point-in-time recovery.

DSNI034I *csect-name* **TABLE SPACE** *table-space-name* **IS IN AUX WARNING STATE**

Explanation: DB2 attempted to recover table space *table-space-name*, but no log records exist because the table space was created with the LOG NO option. The table space is placed in an auxiliary warning state. You can use SQL statements to update the LOB values that are in error or to delete the row that contains the LOB values. In addition, you can use SQL statements to access valid LOBs. However, you cannot use SQL statements to retrieve a LOB value that is not valid.

Severity: 0 (informational)

System action: DB2 marks the LOB values that are in error as not valid. Processing continues.

Programmer response: Use SQL statements to update or delete the LOB values in error. Use the CHECK LOB utility to verify that all errors were corrected and to reset the auxiliary warning state. The auxiliary warning state can also be reset using the REPAIR utility with SET NOAUXWARN.

DSNI036I *csect-name* **THE REAL-TIME STATISTICS COLLECTION PROCESS IS DISABLED**

Explanation: An error prevents DB2 from externalizing the in-memory statistics. The error is generally a persistent error and must be remedied before the process can be enabled. Look for prior messages to identify the nature of the problem.

System action: The statistics database is implicitly stopped and no further attempt will be made to externalize the in-memory statistics.

Operator response: Notify the system programmer.

System programmer response: Inspect the console for prior messages and refer to those messages for problem determination.

DSNI037I *csect-name* **THE REAL-TIME STATISTICS COLLECTION PROCESS COULD NOT ACCESS AN OBJECT IN THE REAL-TIME STATISTICS DATABASE**

Explanation: An error prevents DB2 from externalizing the in-memory statistics. An attempt failed to insert, update, or delete a row in the statistics database. The error is generally transient (lock time-out or resource unavailable). If the operation is retried, it is likely to succeed.

System action: The statistics are kept in-memory until the next opportunity to externalize them.

DSNI038I *csect-name* **THE REAL-TIME STATISTICS COLLECTION PROCESS IS ENABLED**

Explanation: The statistics database started successfully. The required onject descriptors in the statistics database exist and have the correct attributes. DB2 proceeds to externalize in-memory statistics for all modified objects.

Severity: 0 (informational)

System action: The database is started R/w.

DSNI040I *csect-name* **RESTORE SYSTEM UTILITY**
PROCESSING LOG RANGE FROM
rba/lrsn xxxxxx **TO** *yyyyyy*

Explanation: This message periodically shows the
progress of system level recovery made by the

RESTORE SYSTEM utility. In non-data sharing, the log range is RBA. In data sharing, the log range represents LRSN value. The value for *xxxxxxxx* is the RBA or LRSN value of the current log record being processed and the *yyyyyyyy* value is the RBA or LRSN of the last log record to be processed by the RESTORE SYSTEM utility.

Severity: 0 (informational)

System action: System Level Point-In-Time recovery continues.

DSNI041I *csect-name* **START ACCESS(FORCE) CANNOT BE ISSUED AGAINST TARGET** *obj-type obj-qual.obj-name part* **DUE TO STOPPED UTILITIES ON VERSION** *version-number*. **REQUEST IS REJECTED.**

Explanation: The object cannot be started due to outstanding utilities on a different version. The status of the object is unchanged.

version-number is the version on which the utility was running.

obj-type is table space or index space.

obj-qual is the database name.

obj_name is the table space or index space name.

part represents the partition number for a partitioned table space.

System action: The request is rejected.

User response: Use the DISPLAY UTILITY command to see what utilities are outstanding. Issue the -START DATABASE ACCESS(FORCE) command on the same version as the version on which the stopped utility was running. If there is more than one utility outstanding on the object and the utilities were submitted on a different release of DB2, start the correct version of DB2 and use the -TERM UTILITY command to terminate each utility.

DSNI042I *csect-name* **- PAGE SET HEADER RBA** *function*- **ID** *id* **DB ID** *dbid*. **PS ID** *psid* **PART #** *part number* **RBA** *header page* **RBA**

Explanation: This informational message displays the header page RBA that is used to determine the LPL or GRECP recovery range for the specified page set. It is displayed once per LPL recover, per page set, per data sharing group.

System action: The recovery is in progress.

User response: Wait for message DSNI021I, which indicates the end of recovery, or for message DSNI005I, which indicates an error occurred during recovery.

DSNI043I *csect-name* **- MERGED LOG APPLY RANGE** *function*-**ID** *id* **MEMBER ID** *member id* **FROM LRSN/RBA** *lrsn or rba* **TO LRSN/RBA** *lrsn or rba*

Explanation: This informational message displays the broad LRSN or RBA range, merged from all members of the data sharing group. The range is used to determine the LPL or GRECP recovery range for the specified page set. It is displayed once per LPL recovery, per member.

System action: The recovery is in progress.

User response: Wait for message DSNI021I, which indicates the end of recovery, or for message DSNI005I, which indicates an error occurred during recovery.

DSNI044I *csect-name*- **PAGE LOG APPLY RANGE** *function* **-ID** *id* **MEMBER ID** *member id* **FROM PAGE #** *from page number* **TO PAGE #** *from page number* **FROM LRSN/RBA** *LRSN or RBA* **TO LRSN/RBA** *LRSN or RBA*.

Explanation: This informational message displays the LRSN or RBA range that is used to determine LPL or GRECP recovery range for the specified page set in the data sharing group member.

System action: The recovery is in progress.

User response: Wait for message DSNI021I, which indicates the end of recovery, or for message DSNI005I, which indicates an error occurred during recovery.

Chapter 9. DSNJ... messages

DSNJ001I *csect-name* **CURRENT COPY** *n* **ACTIVE LOG DATA SET IS DSNAME=..., STARTRBA=..., ENDRBA=...**

Explanation: This message is generated for one of two reasons:

1. When the log manager is initialized, it sends this informational message to identify the current active log data sets (copy-1 and, if dual logging is used, copy-2).
2. When the current active log data set is full, DB2 will switch to the next available active log data set. This message identifies the next available active log data set which will be used for logging.

The value specified by the STARTRBA=value keyword is the RBA of the first byte of log data in the named data set. The value specified by the ENDRBA=value keyword is the RBA of the last possible byte in the data set. These values may later be needed for a conditional restart.

System programmer response: None required. However, if recovery is required, information from this message may be required as input to the change log inventory (DSNJU003) utility.

DSNJ002I **FULL ACTIVE LOG DATA SET DSNAME=..., STARTRBA=..., ENDRBA=...**

Explanation: This message is sent at the time the log manager fills an active log data set and switches to a new empty data set. The message shows the name and log RBA range of the full data set.

System programmer response: None required. However, if recovery is required information from this message may be required as input to the change log inventory (DSNJU003) utility.

DSNJ003I **csect-name FULL ARCHIVE LOG VOLUME DSNAME=..., STARTRBA=..., ENDRBA=..., STARTxxxx=..., ENDxxxx=..., UNIT=..., COPYnVOL=..., VOLSPAN=..., CATLG=...**

Explanation: Offloading for the specified archive log data set was successfully completed for the given volume. If the data set spans multiple tape volumes, this message is generated for each tape volume. If the data set spans multiple dasd volumes, only one message will be generated displaying the first VOLSER appended with a '+' to indicate the data set spans multiple volumes. 'xxxx' is replaced by either TIME or LRSN.

System action: An archive log data set has been created, and the archive log data set inventory in the bootstrap data set (BSDS) has been updated with the information in the message:

DSNAME
The name of the archive log data set.

STARTRBA
The starting RBA contained in the volume.

ENDRBA
The ending RBA contained in the volume.

STARTLRSN
The starting LRSN contained in the volume for data sharing systems.

ENDLRSN
The ending LRSN contained in the volume for data sharing systems.

STARTTIME
The starting store-clock value of the log records in the volume for non-data sharing systems.

ENDTIME
The ending store-clock value of the log records in the volume for non-data sharing systems.

UNIT
The device unit to which the data set was allocated.

COPYnVOL
The name of the volume. This message keyword is displayed as COPY1VOL if this is the copy-1 archive log data set, and as COPY2VOL if this is the copy-2 archive log data set.

VOLSPAN
An indicator to denote one of four conditions:
- If '00', the data set is entirely contained on the volume specified by 'COPYnVOL'.
- If '01', this is the first entry of a multivolume data set.
- If '11', this is the middle entry of a multivolume data set.
- If '10', this is the last entry of a multivolume data set.

CATLG
An indicator to denote one of two conditions:
- If 'NO', the archive log data set is uncataloged.
- If 'YES', the archive log data set is cataloged.

The BSDS is automatically updated with the information contained in this message; however, if recovery is required, information from this message may be required as input to the change log inventory (DSNJU003) utility.

DSNJ004I **ACTIVE LOG COPY n INACTIVE, LOG IN SINGLE MODE, ENDRBA=...**

© Copyright IBM Corp. 1983, 2008

Explanation: This message is sent when the dual active logging option is selected and copy 'n' becomes inactive. A log copy becomes inactive when the next active log data set is not ready when required. ENDRBA is the last byte of log data written on copy 'n'. This is usually caused by a delay in off-load.

System action: The log is switched to single mode until the next data set for copy 'n' is ready for logging.

If DB2 is shut down or abends while in single mode with the ZPARM option still set for dual active data sets, the previous state of the active log data sets determines what happens when DB2 is started, as follows:

- If fewer than two data sets are available (not flagged as STOPPED) for each set of active logs, DB2 startup terminates and message DSNJ112E is issued.
- If an active log data set is in NOTREUSABLE state, DB2 can be started in single logging mode, but dual mode takes effect when the other active log data set becomes available after offloading.

Operator response: Execute an MVS *Display System Requests* command (D R,L for example) to ensure that there are no outstanding requests that are related to the log off-load process. Take the necessary action to satisfy any requests, and permit off-load to continue.

System programmer response: If the switch to single mode was caused by the lack of a resource required for off-load, the necessary resource should be made available to allow off-load to complete and thus permit dual logging to proceed. If recovery is required, information from this message might be required as input to the change log inventory (DSNJU003) utility.

DSNJ005I ACTIVE LOG COPY n IS ACTIVE, LOG IN DUAL MODE, STARTRBA=...

Explanation: This message is sent when copy 'n' of the log becomes active after previously being flagged as inactive. STARTRBA is the RBA of the first byte of log data written on copy 'n' after it was activated.

System programmer response: None required. However, if recovery is required, information from this message may be required as input to the change log inventory (DSNJU003) utility.

DSNJ006I ALLOCATION FOR NEW ARCHIVE LOG DATA SET HAS BEEN CANCELLED BY OPERATOR.

Explanation: This message is sent if the operator answers 'N' to the WTOR message DSNJ008E.

System action: If the allocation is for copy 1 archive log data set, off-load terminates processing until the next time it is activated. If copy 1 has already been allocated and this request is for copy 2, off-load switches to single off-load mode for this data set only.

DSNJ007I *csect-name* **OPERATOR CANCELED MOUNT OF ARCHIVE VOL SER=** *volser*

Explanation: If the operator answers 'N' to the WTOR message DSNJ009I, this message is issued. The variable 'volser' is the volume serial of an archive log volume required for the read request.

System action: The read request for the archive log volume fails. If the request was issued with the COND=YES parameter, return code 12 and reason code 00D1032B are returned to the requester. Otherwise, DB2 abends with reason code 00D1032B.

DSNJ008E nn OF mm ACTIVE LOGS ARE FULL. ssname NEEDS ARCHIVE SCRATCH. REPLY Y WHEN DEVICE READY OR N TO CANCEL.

Explanation: The installation has requested a WTOR before archive log data set allocation. DB2 needs a scratch volume for offloading an active log data set. 'ssname' is the name of the DB2 subsystem. 'nn' is the number of full active log data sets. 'mm' is the total number of active log data sets.

System action: The off-load task waits for the operator's reply.

Operator response: The operator has three options:

- Get a scratch volume ready, make sure there is an available unit for the volume, and reply 'Y'. DB2 then continues with the off-load.
- Determine from the number of active log data sets available whether the off-load can be delayed until the next time an active log data set becomes full. If the process can be delayed, then reply 'N'.

  This response has two possible effects.

  – If dual archive logging is in effect and this allocation is for a copy 1 archive data set, the 'N' response delays the off-load process until the next active log data set becomes full.

  – However, if the copy 1 archive data set has already been allocated and this request is for copy 2, the 'N' response causes the off-load to switch to single archive mode (the switch is for this data set only).
- Defer giving a response. This causes off-load to wait before processing. However, because off-load is a separate service task, the wait does not affect DB2 performance.

If offloading to DASD, an error has occurred attempting to allocate an archive log data set. Reply 'Y' to receive the error messages.

DSNJ009E ssname NEEDS VOL SER=nnnnnn. REPLY Y TO CONTINUE OR N TO CANCEL

Explanation: The installation requested WTOR before archive log data set allocation. DB2 needs the specified archive volume for a read operation. The subsystem name is 'ssname'.

System action: The archive log read service task waits for the operator's reply. This wait affects the agent for which the log read was issued and any other agents that might be waiting on the log read service task queue.

Operator response: Locate the requested volume, ensure that a device is available, and reply 'Y'. DB2 continues with dynamic allocation and then begins reading the log.

If dual archiving is in effect, a response of 'N' causes archive read to reissue the message for the copy 'Z' copy 2 archive VOLSER with the same RBA range. A response of 'N' to this second WTOR, or to the initial WTOR for single archiving, causes the archive read service task to fail with unpredictable results.

DSNJ010I *csect-name* **INVALID RESPONSE. WTOR WILL BE REPEATED.**

Explanation: A WTOR was issued during archive data set allocation. The user did not respond correctly to the message. Either 'Y' or 'N' must be entered.

System action: The original WTOR is repeated.

Operator response: Reply as indicated in the repeated message.

DSNJ011I CONDITIONAL RESTART RECORD rrr CREATED AT .... WAS FOUND. REPLY Y TO USE, N TO CANCEL

Explanation: When the log manager was initialized, a conditional restart control record was found in the bootstrap data set. Both the record identifier (a 4 byte hexadecimal number) and the creation time stamp (yy.ddd hh:mm) are displayed to help identify the conditional restart record which will be used. If you want a conditional restart using that record, reply 'Y' to the message. Otherwise, reply 'N' and notify the system programmer.

System action: If 'Y' is the response, DB2 is started conditionally, using the record found. If 'N' is the response, DB2 startup is terminated.

Operator response: If a DB2 conditional restart is to be done and if the date and time displayed are correct, then reply 'Y' to the outstanding message. If you are not sure, then contact the system programmer.

System programmer response: If a normal restart of DB2 has failed and you have created a conditional restart control record with the change log inventory (DSNJU003) utility, check whether the time and date in the message agree with when you created that record. If they do, reply 'Y' to the outstanding WTOR. If they do not, reply 'N' and investigate the discrepancy.

DSNJ012I ERROR ccc READING RBA rrr IN DATA SET ddd CONNECTION-ID=xxxxxxxx, CORRELATION-ID=yyyyyyyyyyyy

Explanation: While scanning log records read into a buffer, the log manager detected a logical error with reason code 'ccc'. 'rrr' is the log RBA of the segment in the buffer at which the error was detected. 'ddd' is the name of the active or archive log data set from which the record was read. If 'ddd' is blank, the data was read from an active log output buffer.

The connection and correlation IDs identify the user or application that encountered the problem. Messages that have the same connection ID and correlation ID relate to the same user. Correlation IDs beginning with '0nn' where 'nn' is a number from 01 to 28, identify system agents.

Refer to Part 4 (Volume 1) of *DB2 Administration Guide* for information about recovery from log failures.

System action: The application program that invoked the log manager is terminated with reason code 'ccc'. However, information in this message may be useful in diagnosing the abend that will follow.

DSNJ013I TERMINAL ERROR ccc IN BUFFER rrr BEFORE ACTIVE LOG WRITE

Explanation: A scan of the log output buffer just prior to writing the buffer detected an inconsistency in the log data. 'ccc' is the abend reason code associated with the SDUMP that is produced. 'rrr' is the log RBA at which the error was detected.

System action: DB2 will terminate with a dump, and will not write the damaged buffer to either COPY 1 or COPY 2 active log data set.

Operator response: Notify the system programmer. Start DB2 after it terminates.

System programmer response: Because the damaged buffer has not been written to a log data set, DB2 may be restarted. No corrective action is required.

Problem determination: In the DB2 formatted block section of the dump, locate the chain of LBCE blocks. Those are log buffer control elements, and each LBCE controls one 4KB log output buffer. LBCEBUFF, at offset X'20' in the LBCE, points to the buffer, which is always on a 4KB boundary. The log RBA of the first byte of the buffer is in a 6-byte field at offset X'FF2' in the buffer itself. The layout of the buffer is described by the DSNDLCID mapping macro. The header of each record within the buffer is described by the DSNDLRH mapping macro. Locate the buffer that contains the log

RBA from the message and find the damage by referring to the description of the reason code later in this book.

DSNJ014I TERMINAL ERROR ccc IN BUFFER rrr AFTER ACTIVE LOG WRITE.

Explanation: A scan of the log output buffer, after writing to the COPY 1 active log data set and before writing to COPY 2, detected an inconsistency in the log data. 'ccc' is the abend reason code associated with the SDUMP that is produced. 'rrr' is the log RBA at which the error was detected.

System action: DB2 will terminate with a dump, and will not write the damaged buffer to the COPY 2 data set.

Operator response: Notify the system programmer.

System programmer response: The block containing the indicated log RBA may be damaged. The buffer was found to be in error at the completion of the write to the COPY 1 data set of the active log.

If dual active logs are being used, use print log map (DSNJU004) to list the active log data sets for both copies of the active log. Find the COPY 2 data set with the corresponding RBA, and copy that data set (using access method services REPRO) to the COPY 1 data set. Start DB2.

If only a single active log is used, contact the IBM support center for assistance. An attempt to start DB2 may succeed if the damage to the buffer occurred after completion of the write to DASD.

Problem determination: In the DB2 formatted block section of the dump, locate the chain of LBCE blocks. Those are log buffer control elements, and each LBCE controls one 4KB log output buffer. LBCEBUFF, at offset X'20' in the LBCE points to the buffer, which is always on a 4KB boundary. The log RBA of the first byte of the buffer is in a 6-byte field at offset X'FF2' in the buffer itself. The layout of the buffer is described by the DSNDLCID mapping macro. The header of each record within the buffer is described by the DSNDLRH mapping macro. Locate the buffer that contains the log RBA from the message and find the damage by referring to the description of the reason code later in this book.

DSNJ015I CSRONLY PROHIBITS LOG ALTERATIONS. RECORD rrr CREATED .... REPLY Y TO USE, N TO CANCEL.

Explanation: During DB2 startup, the log manager found an inactive conditional restart control record in the bootstrap data set. Because this control record indicates CSRONLY, no log data sets will be modified. If log truncation was specified, it will be simulated. Only the first phase of restart (current status rebuild) will be performed. The hexadecimal identifier (rrr) and creation timestamp (yy.ddd hh:mm) identify the record found.

System action: A 'Y' reply will continue the conditional restart. A 'N' reply will terminate DB2 startup.

Operator response: To continue this CSRONLY processing, reply 'Y' to this message. If you are not sure, contact the system programmer.

System programmer response: The conditional restart control record will not be deactivated at the end of the CSRONLY processing. To deactivate this record, either cancel this record or create a new record. Refer to Part 4 (Volume 1) of *DB2 Administration Guide* for more information.

DSNJ016E *csect-name* **WARNING - SYSTEM CHECKPOINT PROCESSOR MAY HAVE STALLED. LAST CHECKPOINT WAS TAKEN** *date-time***.**

Explanation: While switching active log data sets, DB2 found that there has not been a system checkpoint taken since the last time DB2 switched active log data sets. *date-time* is the date and time of the last system checkpoint. There should be at least one system checkpoint on each log data set.

System action: DB2 continues processing. If statistics class 3 is active, IFCID 0335 is written. This message will be deleted when a system checkpoint is taken.

System programmer response: This may be an indication that the system checkpoint processor may no longer be functioning. This situation can result in a very long restart if logging continues without a system checkpoint. You should monitor system checkpoint activity very closely. You can use the -DISPLAY LOG command or run the Print Log Map utility (DSNJU004) to display the most recent checkpoint. If DB2 continues logging beyond the defined checkpoint frequency, you should quiesce activity and terminate DB2 to minimize the restart time.

Prior to terminating DB2, obtain a dump of the DB2 address spaces for problem determination.

DSNJ017E *csect-name* **WARNING - OFFLOAD TASK HAS BEEN ACTIVE SINCE** *date-time* **AND MAY HAVE STALLED.**

Explanation: While switching active log data sets, DB2 found that the offload task has been active since the last time DB2 switched active log data sets. The offload task began archiving an active log data set at *date-time*.

System action: DB2 continues processing. If statistics class 3 is active, IFCID 0335 is written. This message will be deleted when the offload task completes the archive process.

System programmer response: This may be an indication that there may be an outstanding tape

mount request or that the offload task may no longer be functioning. This situation can result in a log-full situation if DB2 can not archive log data sets when they fill. You should reply to any tape mount requests and monitor offload activity very closely. You can issue the -DISPLAY LOG command to display the status of the offload task, and the -ARCHIVE LOG CANCEL OFFLOAD command can be used to terminate and restart the offload process.

Prior to terminating the offload task, obtain a dump of the DB2 MSTR address space for problem determination.

DSNJ020I *csect-name* **RECEIVED REPLY OF N TO** *msg-num***. SUBSYSTEM STARTUP IS TERMINATED.**

Explanation: The operator chose to terminate DB2 startup by answering 'N' to 'msg-num'.

System action: DB2 will not restart.

Operator response: To restart DB2, follow the 'operator response' of 'msg-num'.

DSNJ030I RBA RANGE startrba TO endrba NOT AVAILABLE IN ACTIVE LOG DATA SETS

Explanation: Previous errors have made the active log data set(s) that contain the relative byte address (RBA) range reported in the message unavailable to DB2. The status of these log(s) is STOPPED in the bootstrap data set (BSDS).

System action: DB2 terminates with a dump.

Operator response: Notify the system programmer.

System programmer response: The log RBA range must be available for DB2 to be recoverable. Correct previous errors and restore the active log data set(s) that contain the RBA range reported in the message.

- If the log data set(s) are recoverable, the active log data set inventory in the BSDS must be modified to reset the STOPPED status. Use the print log map (DSNJU004) utility to obtain a copy of the BSDS log inventory. Next, use the change log inventory (DSNJU003) utility to delete the active log data sets marked STOPPED (use the DELETE statement), and then add them again (use the NEWLOG statement). The starting and ending RBA for each active log data set must be specified on the NEWLOG statement when the log(s) are added back to the BSDS using the DSNJU003 utility.
- If the log data set(s) are not recoverable, it is necessary to conditionally restart DB2. See Part 4 (Volume 1) of *DB2 Administration Guide* for additional information on conditional restart and recovering from log failures.

Problem determination: Examine previous messages to determine the reason the active log data set(s) are unavailable to DB2.

DSNJ031I *csect-name* **WARNING - UNCOMMITTED UR HAS WRITTEN** *nn* **LOG RECORDS - CORRELATION NAME =** *xxxxxxxxxxxx* **CONNECTION ID =** *yyyyyyyy* **LUWID = LOGICAL-UNIT-OF-WORK-ID=***token* **PLAN NAME =** *xxxxxxxx* **AUTHID =** *xxxxxxxx* **END USER ID =** *xxxxxxxx* **TRANSACTION NAME =** *xxxxxxxx* **WORKSTATION NAME =** *xxxxxxxx*

Explanation: This message indicates that a unit of recovery (UR) reached the threshold number of log records written without a COMMIT or ROLLBACK. This threshold was defined during installation.

The value *nn* is cumulative, and indicates the number of log records that were written since the beginning of the UR.

CORRELATION name, CONNECTION ID, and LUWID together identify a thread associated with the UR. *token* is a unique token number associated with the LUWID. If the LUWID is an '*', it indicates that the thread originated at this site.

PLAN NAME and AUTHID further identify the thread associated with the UR.

If the thread's creator included End User information, then DB2 displays the End User's ID, TRANSACTION NAME, and WORKSTATION NAME. Otherwise, these fields will contain an '*'.

System action: DB2 continues processing. If statistics class 3 is active, IFCID 0313 is written. This message will be repeated each additional time the threshold is reached.

System programmer response: Consult with the application programmer to determine if this is a problem UR. See Part 4 (Volume 1) of *DB2 Administration Guide* for more information about problems caused by uncommitted UR's.

If an application program caused the UR to reach its threshold number of log records, you can use CANCEL THREAD to cancel the UR. If you cancel an uncommitted UR, DB2 will undo the changes. The amount of time required for this process depends on the amount of work done by the UR. The amount of time required for this process depends on the amount of work done by the UR.

Programmer response: Ensure that the application commits frequently enough, or consult with your DB2 administrator about decreasing the frequency of the check.

DSNJ032I *csect-name alert-lvl*- **APPROACHING END OF THE LOG RBA RANGE OF** *FFFFFFFFFFFF*. **CURRENT LOG RBA IS** *xxxxxxxxxxxx*

Explanation: The current log RBA is approaching the end of the log RBA range. *xxxxxxxxxxxx* is the current log RBA value. The current log RBA should not be allowed to advance to the maximum value of *FFFFFFFFFFFF*. *alert-lvl* indicates one of the following:

- WARNING is issued when the current log RBA reaches the *F80000000000* value. DB2 will continue processing.
- CRITICAL is issued after the log RBA value reaches *FFFF00000000*. If the message is issued during DB2 restart, then processing will continue. Otherwise, DB2 will terminate with reason code 00D10251

System action: DB2 continues processing if the *alert-lvl* is WARNING or if DB2 is restarting. Otherwise, DB2 terminates. Any attempt to restart DB2 in general access mode following the CRITICAL termination will result in DB2 restarting in "light" mode, and DB2 will shut down after restart completes the recovery of the incomplete units-of-recovery. DB2 can then be restarted in ACCESS(MAINT) mode, in order to recover any indoubt units-of-recovery, to -START any table spaces in a restricted state such as LPL or GRECP, or to -TERMINATE any utilities that are in progress.

Operator response: Notify the system programmer.

System programmer response: Contact IBM Support for assistance and information on current procedures on how to quiesce DB2 and reset the log RBA value.

DSNJ050I **LOAD MODULE DSNJL003 DOES NOT HAVE RMODE(24) ATTRIBUTE**

Explanation: The named load module was not link edited with the required RMODE(24) attribute.

System action: DB2 startup is terminated.

Operator response: Notify the system programmer of the failure.

System programmer response: Verify that all installation and maintenance activities against this module were executed using SMP. Verify that the JCLIN for SMP includes the correct RMODE control statement, and that the Linkage Editor provided with the Data Facility Product (DFP) for the version of MVS being used was invoked by SMP for the link edits. Correct the procedure or JCL which caused the error and rerun the necessary SMP jobs.

Problem determination: The MVS service aid AMBLIST provides Linkage Editor module attributes in the summary section of the LISTLOAD function output.

DSNJ070I *csect-name* **ARCHIVE LOG DSN PREFIX NOT IN PROPER FORMAT TO RECEIVE TIMESTAMP DATA. TIMESTAMPING OF** *dsnname* **BYPASSED**

Explanation: By way of a ZPARM parameter, the user has specified that the date and time of creation of an archive log data set be included as part of the archive log data set name (DSN). To accomplish this, the DB2 subsystem requires that the length of the archive log data set name prefix not exceed 19 characters if using the standard DATE formate (YYDDD), or 17 characters if using the extended DATE format (YYYYDDD). If the prefix requirement is not met, this message is issued just prior to the allocation of the archive log data set specified in the message.

System action: The archive log data set will be allocated using the archive log prefix. However, the archive log DSN will not contain the date and time as the user requested.

Operator response: Notify the system programmer.

System programmer response: The ZPARM parameters for the log archive function must be changed. Specifically, the TSTAMP and ARCPFXn fields are not consistent with one another. For details concerning the course of action required to eliminate this problem, refer to Part 2 of *DB2 Installation Guide*, under the DSNTIPH panel and DSN6ARVP macro subtopics.

DSNJ071I *csect-name* **TIMER FAILURE CAUSED TIMESTAMPING OF ARCHIVE** *dsname* **TO BE BYPASSED**

Explanation: By way of a ZPARM parameter, the user has specified that the date and time of creation of an archive log data set be included as part of the archive log data set name (DSN). However an attempt to get the current date and time from the system failed. This message is issued just prior to the allocation of the archive log data set specified in the message.

System action: The archive log data set will be allocated using the archive log prefix. However, the archive log DSN will not contain the date and time as the user requested.

DSNJ072E **ARCHIVE LOG DATA SET dsname HAS BEEN ALLOCATED TO NON-TAPE DEVICE AND CATALOGED. ZPARM CATALOG OPTION OF 'NO' HAS BEEN OVERRIDDEN.**

Explanation: By way of an archive log DSNZPARM parameter, the user specified that all archive log data sets should be uncataloged (CATLG=NO). However, DB2 requires that all archive log data sets allocated to nontape devices must be cataloged. The archive log

data set specified by 'dsname' has been allocated to a nontape device, and has thus been cataloged. The user's DSNZPARM CATLG setting of 'NO' has been overridden.

System action: The archive log data set has been allocated to a nontape device, and has been cataloged. The DSNZPARM CATLG=NO setting has been overridden. The BSDS reflects that the data set has been cataloged.

Operator response: Notify the system programmer.

System programmer response: The DSNZPARM parameters for the log archive function must be changed. Specifically, the ARCHCTLG and ARCHDEVT fields are not consistent with one another. For details concerning the course of action required to eliminate this problem, refer to Part 2 of *DB2 Installation Guide* under the DSNTIPA panel and DSN6ARVP macro subtopics.

DSNJ073I LOG ARCHIVE UNIT ALLOCATION FAILURE DETECTED, RETURN CODE=nnnn. ALLOCATION OR OFF-LOAD OF ARCHIVE LOG DATA SET MAY FAIL.

Explanation: While building the SVC99 text entries to dynamically allocate a new archive log data set, a unit allocation failure was detected. This message is only generated when a trace for IFCID 320 is active . The reason code, indicated by 'nnnn' in the message, further clarifies the problem as follows:

4 (X'04')
The device type (unit name) on which the archive log data set is to be allocated is specified by the user in the DB2 system parameters. No UCBs could be found for the device type (unit name).

8 (X'08')
The device type (unit name) on which the archive log data set is to be allocated is specified by the user in the DB2 system parameters. This device type (unit name) could not be found in the MVS eligible device table (EDT).

32 (X'20')
DB2 was able to obtain a list of devices corresponding to the device type (unit name) specified in ZPARMs. However, it was determined by DB2 that this list contained a mixture of tape and nontape devices.

40 (X'28')
The device type (unit name) specified by the user in ZPARMs is valid. However, no devices are currently associated with the given device type (unit name).

44 (X'2C')
The device type (unit name) specified by the user in ZPARMs is valid. However, no DASD volumes are available with a volume use attribute of 'storage'.

48 (X'30')
The CVT indicates that UCB services are not available.

System action: This message is issued by DB2 after the SVC99 text entries are built, but prior to the allocation of the new archive log data set. As a result of the error, the dynamic allocation of the archive log data set will be attempted using standard default values. The standard default values are generally acceptable; however, the allocation may fail or the subsequent off-load may produce undesirable processing results. For example:

- A return code of 4 or 44 (X'2C') will foretell an allocation error (DSNJ103I) when the SVC99 is issued for the archive data set.
- Off-load processing to tape may fail. DB2 uses a volume count of 20 when allocating to tape, and uses the standard MVS volume count default of 5 volumes when writing to nontape devices. In the case of most of the above errors, it would be impossible for the DB2 subsystem to determine the device type on which the data set is to be allocated. Therefore, the standard MVS default is assumed for the volume count. If the data set is successfully allocated to a tape device, and the volume of data is such that more than five volumes will be used for the archive data set, the off-load processing will receive an MVS ABEND 837-08 with message IEC028I when attempting to write to the sixth tape volume.
- Off-load processing to a direct access device may fail. When allocating a new archive log data set on a direct access device, the DB2 subsystem will use a unit count to facilitate multivolume archive data sets. With most of the above errors, it may be impossible for the DB2 subsystem to correctly determine the type of device on which the data set is to be allocated. Therefore, the standard default (1) is assumed for the unit count. If the data set is successfully allocated to a direct access device, and during the off-load processing it becomes necessary to extend the data set to another device, the off-load processing will receive an MVS B37 (out of space) ABEND, and the archive log data set will be deallocated.

Operator response: Notify the system programmer.

System programmer response: The required action is based on the return code indicated in the message.

4 (X'04') or 8 (X'08')
The most likely cause is an incorrect specification in the archive log ZPARMs (DSNTIPA panel). Correct the DEVICE TYPE field. If the DEVICE TYPE field from the archive log ZPARMs appears to be correct, check the EDT to ensure that the esoteric or

generic unit name specified in the ZPARMs is actually in the EDT. Subsequent off-load processing will archive the log data which could not be previously archived due to the allocation failure (DSNJ103I).

32 (X'20') or 40 (X'28')
To correct this situation, change the archive log DEVICE TYPE ZPARM to use a device type (unit name) which contains homogenous devices, or modify the device list associated with the device type (unit name) via a system generation to supply a list of homogenous devices.

44 (X'2C')
To correct this situation, issue an MVS mount command to change the volume use attribute of a mounted 'private' volume to 'storage'. If this is a recurring problem, a system generation may be required to add permanently resident volumes with a volume use attribute of 'storage' to the esoteric or generic unit, or ZPARMs may have to be changed to use a different esoteric or generic unit name for the DEVICE TYPE (on the DSNTIPA installation panel).

48 (X'30')
UCB services became available in MVS/ESA Version 4.1 (HBB4410).

DSNJ096E *csect-name* **CRCR ENDLRSN RBA VALUE OF** *rrr* **IS INVALID, NO TRUNCATION WILL OCCUR**

Explanation: Truncation cannot take place because the RBA value specified on the active CRCR record is not equal to the RBA of a log record. *rrr* is the ENDLRSN specified in the active record. The ENDLRSN RBA value must match the start of a log record in a non-data sharing environment.

System action: DB2 startup is terminated.

Operator response: Notify the system programmer.

System programmer response: Confirm that the ENDLRSN value specified in the Conditional Restart control record is correct. Run the Change Log Inventory (DSNJU003) utility with the CRESTART CANCEL parameters to cancel the conditional restart. If log truncation is desired, run the DSNJU003 utility with the correct ENDLRSN value. Restart DB2.

DSNJ098E *csect-name* **CRCR ENDLRSN OF** *rrr* **IS** *xxx* **THAN ANY KNOWN** *yyyy***, NO TRUNCATION WILL OCCUR**

Explanation: Truncation cannot take place because the CRCR ENDLRSN is not in the range of LRSN or RBA values known to either the active or archive logs. *rrr* is the end LRSN specified in the active record. *xxx* is replaced by either HIGHER or LOWER. In a data sharing environment, if *xxx* is HIGHER, the CRCR ENDLRSN is higher than the LRSN of the last record written to the log data sets. If *xxx* is LOWER, the CRCR ENDLRSN is lower than the starting LRSN of the oldest archive data set. *yyyy* is replaced by LRSN. In a non-data sharing environment, if *xxx* is HIGHER, the CRCR ENDLRSN is higher than the RBA of the last record written to the log data sets. If *xxx* is LOWER, the CRCR ENDLRSN value is an RBA value that is lower than the starting LRSN of the oldest archive data set. *yyyy* is replaced by RBA.

System action: DB2 startup is terminated.

Operator response: Notify the system programmer.

System programmer response: Confirm that the ENDLRSN value specified in the Conditional Restart control record is correct. If it is not correct, run the Change Log Inventory (DSNJU003) utility with the CRESTART CANCEL parameters to cancel the conditional restart. If log truncation is desired, run the the DSNJU003 utility with the correct ENDLRSN. Restart DB2.

DSNJ099I LOG RECORDING TO COMMENCE WITH STARTRBA=...

Explanation: This message is generated when the log manager is initialized during DB2 startup. The value specified by the STARTRBA=value keyword is the RBA of the next byte of log data to be recorded in the DB2 active log data set(s). This information may later be needed for a conditional restart.

This message is preceded by one DSNJ001I message (if single logging) or two DSNJ001I messages (if dual logging).

System programmer response: None required. However, if recovery is required, information from this message may be required as input to the change log inventory (DSNJU003) utility.

DSNJ100I *csect-name* **ERROR OPENING BSDSn DSNAME=..., ERROR STATUS=***eeii*

Explanation: Log manager initialization or the -RECOVER BSDS command could not open the specified bootstrap data set (BSDS). BSDSn matches the ddname in the log startup JCL of the data set that cannot be opened. The value of 'n' is 1 or 2. Refer to Part 4 (Volume 1) of *DB2 Administration Guide* for information about recovery from BSDS or log failures.

System action: When this error occurs at initialization time, DB2 startup must be terminated, because the log data sets cannot be determined and allocated without the BSDS. When this error occurs during -RECOVER BSDS processing, the command is terminated, and the system continues in single BSDS mode.

Operator response: Notify the system programmer.

System programmer response: Recover the BSDS that

cannot be opened. Normally, if dual BSDSs are being used, recovery consists of making a copy of the remaining good BSDS.

Problem determination: The 'error status' contains the VSAM open return code in 'ee', and the VSAM open reason code in 'ii'. Refer to *z/OS DFSMS: Macro Instructions for Data Sets* for a list of the VSAM open return codes and reason codes, and the steps required to take corrective action.

DSNJ101E *csect-name* **CRCRERBA OF** *rrr* **IS** *xxx* **THAN ANY KNOWN RBA, BUT COLD START WAS NOT SPECIFIED**

Explanation: Truncation cannot take place because the CRCR end RBA is not in the range of RBA values known to either the active or archive logs. 'rrr' is the end RBA specified in the active record. 'xxx' is replaced by either HIGHER or LOWER. If HIGHER is the case, the CRCRERBA is higher than the end RBA of the most recent active log data set. LOWER means the CRCRERBA is lower than the starting RBA of the oldest archive data set.

System action: DB2 startup is terminated.

Operator response: Notify the system programmer.

System programmer response: Most likely, the archive log data set that contained the requested RBA has been deleted from the bootstrap data set (BSDS) by the change log inventory (DSNJU003) utility. Locate the output from an old Print Map run, and identify the data set that contains the missing RBA. If the data set has not been reused, run the DSNJU003 utility to add this data set back into the inventory of log data sets. Restart DB2.

DSNJ102I LOG RBA CONTENT OF LOG DATA SET DSNAME=..., STARTRBA=..., ENDRBA=..., DOES NOT AGREE WITH BSDS INFORMATION

Explanation: The log RBA range shown in the BSDS for the specified data set does not agree with the content of the data set. This message indicates that the change log inventory (DSNJU003) utility was used incorrectly or that a down-level volume is being used.

System action: DB2 startup processing is terminated.

Operator response: Notify the system programmer.

System programmer response: Use the print log map (DSNJU004) and change log inventory (DSNJU003) utilities to make the BSDS consistent with the log data sets.

DSNJ103I *csect-name* **LOG ALLOCATION ERROR DSNAME=***dsname***, ERROR STATUS=***eeeeiiii***, SMS REASON CODE=***ssssssss*

Explanation: The log manager encountered an error while attempting to allocate the active or archive log data set indicated by the DSNAME=keyword. The STATUS=keyword indicates the error reason code returned by MVS dynamic allocation (SVC99).

This message might be preceded by message DSNJ073I.

System action: Subsequent log manager actions are based upon the type of data set involved.

For active log data sets, if the error is encountered during log manager initialization, DB2 startup is terminated. If two copies of the active log data sets are defined, this message appears only once.

For archive log data sets, if two copies of the archive log data sets are defined, processing continues on the remaining archive log data set.

Operator response: Notify the system programmer.

System programmer response: The error status portion of this message contains a 2-byte error code (S99ERROR) followed by the 2-byte information code (S99INFO) from the SVC99 request block. If the S99ERROR code indicates an SMS allocation error (97XX), then 'ssssssss' contains additional SMS reason code information obtained from S99ERSN. Refer to *z/OS MVS Programming: Authorized Assembler Services Guide* for a description of the SVC99 codes. Additional SMS reason codes can be found in *z/OS DFSMSdfp Diagnosis Reference*.

For active log data sets, if the problem occurred during DB2 initialization, you can resolve the problem by either:

- Resolving the error associated with the active log data set as indicated by STATUS=keyword
- Using AMS, providing another copy of the active log data set
- Updating the BSDS with the change log inventory (DSNJU003) utility
- Restarting DB2.

For archive log data sets:

- If the problem occurred during allocation with the intent to write the data set, no immediate action is required. However, failure to resolve the SVC99 error (indicated by the STATUS value in the message) might eventually cause the available space in the active log to be exhausted (DSNJ111E), since all future off-loads might fail due to the same error.
- If the problem occurred during allocation with the intent to read the data set, determine the problem, and use the DSNJU003 utility DELETE function to delete the archive log data set from the BSDS archive log inventory. Then use the NEWLOG function to

add the data set back into the archive log inventory, pointing to the correct volume and device.

If additional information is required about recovery from log failures, refer to Part 4 (Volume 1) of *DB2 Administration Guide*.

This message can also be issued as the result of a possible user error. If the STATUS=keyword displays a value of '17080000', then the site might have one or more active log data sets defined in the BSDS, but not allocated on DASD. To correct the situation, print the contents of the current active log data set inventory using the print log map (DSNJU004) utility, and then do *one* of the following:

- For each active log data set listed in the BSDS, but not actually allocated on DASD, allocate the active log data set using access method services. The access method services command syntax for active log data sets can be found in the DSNTIJIN installation job stream.
- Use the DSNJU003 utility DELETE statement to delete the errant active log data set name, and the NEWLOG statement to add the correct name to the active log data set inventory. The name specified on the NEWLOG statement must be the same as the name of the actual DB2 active log data set allocated on DASD.

DSNJ104I *csect-name* **RECEIVED ERROR STATUS** *nnn* **FROM** *macro-name* **FOR DSNAME** *dsname*

Explanation: DB2 encountered an error while issuing macro 'macro-name'. Error status is the return code from the specified macro.

- For an OPEN of a VSAM data set, the return code in the error field of the access method services control block is included in this message as the error status value. Refer to the appropriate VSAM publication for a definition of these values.
- If the OPEN was for a non-VSAM data set, the error status is zero.
- For MMSRV errors, error status contains the error information returned by media manager services. If an MMSRV CATUPDT failure occurs attempting to truncate an active log data set, the log data set will be unavailable to DB2 and the status of the log data set will be flagged as 'STOPPED' in the BSDS.
- For VSAM OPEN and MMSRV errors, this message is preceded by an IEC161I message that defines the error that occurred.
- For a PROTECT of an archive log data set, the return code is from DADSM PROTECT. Refer to the appropriate MVS publication for a description of the MMSRV return codes.
- For a CLOSE of an archive log data set, the return code is the TCB completion code for an abend received during CLOSE processing. This abend is normally preceded by an IEC210I or IEC212I message that defines the error.
- For EDTINFO and UCBLOOK errors, the first 2 bytes contain the return code and the second 2 bytes contain the reason code.

Refer to Part 4 (Volume 1) of *DB2 Administration Guide* for information about recovery from log failures.

System action: If this condition is encountered during log manager initialization, DB2 startup is terminated. If the error occurs later and the data set is needed either for offload or for input operations, DB2 processing continues. If a second copy of the data is available, DB2 attempts to allocate and open the second data set.

If the data set is needed as an active log data set, the log manager attempts to retry the failing request. If the retry fails, the DB2 subsystem is terminated.

Operator response: Notify the system programmer.

System programmer response: If the error occurred during initialization, the installation must either correct the problem so that the data set is available or provide another copy of the data set and change the BSDSs to point to the new data set.

If the error occurred during DB2 processing, the return code should be reviewed and the appropriate action taken to correct the problem so that the data set can be used at a later time or the data set entry can be removed from the BSDS via the change log inventory (DSNJU003) utility.

If the error was received from PROTECT, there may be a problem with the PASSWORD data set. You should refer to the appropriate DADSM publication to determine the cause of the problem. When the problem has been corrected, you should ensure the archive log data sets receiving the error are added to the PASSWORD data set. If these archive log data sets are not added to the PASSWORD data set, archive read will fail when attempting to OPEN these data sets.

If the error was received from EDTINFO or UCBLOOK and the allocation for the archive log data set fails, verify that the archive UNIT names are correct in ZPARMs.

DSNJ105I *csect-name* **LOG WRITE ERROR DSNAME=..., LOGRBA=..., ERROR STATUS=***ccccffss*

Explanation: The log manager encountered a write error in the specified active log data set. If 'csect-name' is DSNJW107, the error occurred writing the log buffers to an active log data set. If 'csect-name' is DSNWJ207, the error occurred while preformatting the next control area before writing log data into it.

Error status contains the error information returned by media manager in the form 'ccccffss', where 'cccc' is a 2-byte return code that describes the error, 'ff' is a

1-byte code that defines the functional routine that detected the error, and 'ss' is the 1-byte status code that defines a general category of error. Refer to the appropriate MVS publication for a description of the MMCALL return codes.

If the dual active logging option is selected, the log manager switches to the next data set for this copy. If the next data set is not ready, the log manager temporarily enters single logging mode and allocates a replacement data set for the one that encountered the error. Dual logging is resumed as soon as possible.

If single active logging option is selected and the next data set is not ready, the log manager waits for that data set to be available. In this case, log writing is inhibited until the replacement is ready for output.

System action: See the Explanation section of this message.

Operator response: Notify the system programmer.

System programmer response: See the Explanation section of this message.

DSNJ106I LOG READ ERROR DSNAME=..., LOGRBA=..., ERROR STATUS=ccccffss

Explanation: The log manager encountered an error while reading an active log data set. The 'error status' contains the error information returned by the media manager in the form 'ccccffss', where 'cccc' is a 2-byte return code that describes the error, 'ff' is a 1-byte code that defines the functional routine that detected the error, and 'ss' is the 1-byte status code that defines a general category of error.

System action: If another log data set contains the data, the log manager attempts to read the data from the alternate source. If an alternate source is not available, the log manager returns a read error return code to the program requesting the log data.

Operator response: Notify the system programmer.

System programmer response: If the site is using dual logging, the requested RBA was probably retrieved from the corresponding dual active log data set, and no immediate response is necessary. However, if this error occurs frequently, or if the site is using single logging, immediate attention may be required.

Since this condition might indicate an error detected by the media manager, refer to *z/OS DFSMSdfp Diagnosis Reference* for a description of media manager return codes. Take corrective action based on these codes.

It might be necessary to replace the bad data set with a new data set containing the log data, and to update the BSDSs to reflect the new data set using the change log inventory (DSNJU003) NEWLOG operation.

For additional information about recovery from log failures, refer to Part 4 (Volume 1) of *DB2 Administration Guide*

This message can also be issued as the result of a possible user error. If the data set name associated with the DSNAME=keyword is missing, and the STATUS=keyword displays a value of '00180408' or '00100408', then the site is using dual logging, but only one set of active log data sets are defined in the BSDS. To resolve this condition, do *one* of the following:

- Define a second set of active log data sets using access method services (if they are not defined already), and update the BSDS log inventory using the DSNJU003 NEWLOG operation.
- Reset the log parameters ZPARM to indicate single logging. To do this, set the TWOACTV parameter in the DSN6LOGP ZPARMs to NO.

DSNJ107I READ ERROR ON BSDS DSNAME=*name* ERROR STATUS=*status*

Explanation: The log manager encountered an error while reading the specified bootstrap data set. Error Status contains the VSAM return and feedback codes. It is a 2-byte field with the first byte containing the hexadecimal return code and the second containing the hexadecimal feedback code. Refer to the appropriate MVS publication for a description of VSAM return and reason codes.

Refer to Part 4 (Volume 1) of *DB2 Administration Guide* for information about recovery from BSDS or log failures.

System action: If dual BSDSs are available, the log manager attempts to read from the other BSDS. If read from the second BSDS fails or if there is only one BSDS, an error code is returned to the log request that caused access to the BSDS.

If read error is detected during start-up, DB2 initialization process is terminated to prevent users from operating in a cripple mode. User should take action to correct the problem.

Operator response: Notify the system programmer.

System programmer response: It may be necessary to replace or repair the BSDS, depending on what conditions resulted from the read error. To replace a BSDS, first delete the bad BSDS, and then define the new BSDS with the same name and attributes. If a new name is used for the new BSDS, change the log startup JCL to specify the new BSDS name.

DSNJ108I WRITE ERROR ON BSDS DSNAME=... ERROR STATUS=...

Explanation: The log manager encountered an error while writing to the specified bootstrap data set. Error Status contains the VSAM return and feedback codes. It is a 2-byte field with the first containing the hexadecimal return code and the second containing the hexadecimal feedback code. Refer to the appropriate VSAM publication for a description of VSAM return and reason codes.

System action: If dual BSDSs are available, the log manager enters single BSDS mode using the remaining good BSDS. Otherwise, an error code is returned to the log request that caused access to the BSDS.

Operator response: Notify the system programmer.

System programmer response: If dual BSDS mode is being used, execute an offline access method services job to rename the error BSDS and define a new BSDS with the same name. Then enter the log manager -RECOVER BSDS command to reestablish dual BSDS mode.

If dual BSDS mode is not being used, the DB2 subsystem must be shut down, and the BSDS must be recovered from a backup copy. To recover the BSDS, use the change log inventory (DSNJU003) utility.

DSNJ109I OUT OF SPACE IN BSDS DSNAME=...

Explanation: There is no more space in the specified bootstrap data set. The operation that encountered the out-of-space condition did not complete properly.

System action: If dual BSDSs are available, the log manager enters single BSDS mode using the remaining good BSDS. Otherwise, an error code is returned to the log request that caused access to the BSDS.

Operator response: Notify the system programmer.

System programmer response: If dual BSDS mode is being used, execute an offline access method services job to rename the full BSDS and define a new, larger BSDS with the same name. Enter the DB2 -RECOVER BSDS command to reestablish dual BSDS mode.

If dual BSDS mode is not being used, the DB2 subsystem must be shut down and the BSDS recovered offline. In this case, execute the same access method services job mentioned above to rename the full data set and define a larger data set. Next, execute an access method services REPRO job to copy the full BSDS into the new BSDS.

DSNJ110E LAST COPYn ACTIVE LOG DATA SET IS nnn PERCENT FULL

Explanation: This message is issued when the last available active log data set is 5% full, and is reissued after each additional 5% of the data set space is filled.

System action: Each time the message is issued, the off-load processing is retried. If the situation is not corrected, the active log data space fills to capacity, message DSNJ111E is issued, and the DB2 online processing halts.

Operator response: To clear this condition, steps must be taken to complete other waiting off-load tasks. Once an active log data set is made available (reusable) by completing the off-load process for it, the DB2 logging activity can continue.

Execute an MVS *Display System Requests* command (D R,L for example) to determine the outstanding requests related to the log off-load process. Take the necessary action to satisfy any requests, and permit off-load to continue. If off-load fails to complete normally or cannot be initiated, notify the system programmer.

System programmer response: Either correct the problem that is causing the off-load process to fail, or add more active log data sets. Note that the latter action requires DB2 to be down and the change log inventory (DSNJU003) utility to be run.

Possible causes for the shortage of active log data space are:

- Excessive logging. For example, too many users have requested logs at the same time or too many utilities have specified LOG(YES).
- Delayed or slow off-loading. For example, failure to mount archive volumes, incorrect replies to off-load messages, or slow device speeds.
- Excessive use of the -ARCHIVE LOG command. Each invocation of the -ARCHIVE LOG command causes the DB2 subsystem to switch to a new active log data set and to initiate an off-load of the active log. Although the -ARCHIVE LOG command will not be processed when only one active log data set remains in a copy of the active log (see DSNJ319I), excessive use of the command could have consumed all space in the active log except the current active log data sets.
- Off-loads failed. See Operator Response.
- Insufficient active log space. See Operator Response.
- Offload task no longer functioning. Issue the -ARCHIVE LOG CANCEL OFFLOAD command to abnormally terminate and restart the offload process.

DSNJ111E OUT OF SPACE IN ACTIVE LOG DATA SETS

Explanation: Due to delays in off-load processing, all available space in all active log data sets has been exhausted. DB2 recovery logging cannot continue.

System action: The DB2 log manager waits for an available data set.

Operator response: Execute an MVS *Display System Requests* command (D R,L for example) to ensure that there are no outstanding requests that are related to the log off-load process. Take the necessary action to satisfy any requests, and permit off-load to continue.

System programmer response: If the delay was caused by the lack of a resource required for off-load, the necessary resource must be made available to allow off-load to complete and thus permit logging to proceed. For additional information on recovery from this condition, refer to the discussion on DB2 system resource failures and active log failures in Part 4 (Volume 1) of *DB2 Administration Guide*.

DSNJ112E *csect-name* **INSUFFICIENT ACTIVE LOG DATA SETS DEFINED IN BSDS**

Explanation: There are not enough active log data sets defined in the BSDS to start DB2. This condition usually exists for one of the following reasons.

- Fewer than two data sets are defined for one of the active log copy sets.
- DSN6LOGP specified TWOACTV=YES in DB2 initialization parameters, but data sets for two copies of active log are not defined in BSDS.
- Fewer than two data sets are available (not flagged as STOPPED) for one of the active log copy sets.

System action: DB2 startup is terminated.

Operator response: Notify the system programmer.

System programmer response: Use the change log inventory (DSNJU003) utility to make the number of active log data sets defined in the BSDS consistent with the parameters specified on DSN6LOGP. Log data sets that are flagged as 'STOPPED' will not be reused by DB2. These data sets must be recovered (either redefined or recovered from the other copy of the log) and the log data sets must be deleted and re-added to the BSDS (using DSNJU003) with the appropriate RBA's, to clear the STOPPED status. Restart DB2.

DSNJ113E **csect-name** *log-type log-value* **NOT IN ANY ACTIVE OR ARCHIVE LOG DATA SET. CONNECTION-ID=***aaaaaaaa***, CORRELATION-ID=***aaaaaaaa***, MEMBER-ID=***member-id***

Explanation: A request was made to read the log starting at this *log-value* and *log-type* (RBA or LRSN). However, the log value cannot be found in any active or archive log data set for the indicated member. The *member-id* contains the ID of the data sharing member whose logs were being requested at the time of the failure or zero if data sharing is not active. The connection and correlation IDs identify the user or application that encountered the problem. Correlation IDs beginning with '0nn' where 'nn' is a number from 01 to 28, identify system agents. Refer to Part 4 (Volume 1) of *DB2 Administration Guide* for information about recovery from log failures.

System action: Depending upon what log record is being read and why, this error could result in an abend with a reason code of '00D1032A'.

Operator response: Notify the system programmer.

System programmer response: This error could be a user error. In most cases, the archive log data set that contained the requested log value has been deleted from the bootstrap data set (BSDS) by the change log inventory (DSNJU003) utility. Complete the following steps to correct the error:

- Locate the output from an old print log map run of the BSDS for the DB2 member identified by *member-id*.
- Identify the data set that contains the missing log value. If the data set has not been reused, run the DSNJU003 utility to add this data set back into the inventory of log data sets.
- Restart DB2.

DSNJ114I **ERROR ON ARCHIVE DATA SET, OFFLOAD CONTINUING WITH ONLY ONE ARCHIVE DATA SET BEING GENERATED**

Explanation: An error occurred while accessing one of the archive data sets being created by off-load. Because the dual archive option is specified, off-load is continuing with the other archive data set. For the RBA range being off-loaded, there is only one copy of archive instead of the usual two copies.

System action: Off-load produces a single archive data set.

System programmer response: A second copy of this archive log data set can be made, and the BSDS(s) can be updated with the change log inventory (DSNJU003) utility.

DSNJ115I **OFFLOAD FAILED, COULD NOT ALLOCATE AN ARCHIVE DATA SET**

Explanation: Off-load could not allocate or open an archive log data set. The off-load was not performed. If this message is preceded by message DSNJ103I, the off-load data set could not be allocated. If this message is not preceded by DSNJ103I, the off-load data set was successfully allocated, but could not be opened.

System action: Off-load will be tried at a later time.

Operator response: Notify the system programmer.

System programmer response: If the allocation failed with DSNJ103I, review the error status information of message DSNJ103I. Correct the condition that caused the data set allocation to fail so that, on retry, the off-load can take place.

DSNJ116I **ERROR ATTEMPTING TO ADD ARCHIVE ENTRY TO BSDS**

Explanation: Off-load could not add an archive entry to the BSDS. The off-load is considered incomplete. The active log data set is not marked as reusable for new log data. This message is preceded by message DSNJ107I, DSNJ108I, or DSNJ109I.

System action: Off-load will be retried at a later time.

Operator response: Notify the system programmer.

System programmer response: Refer to the specific preceding message for action.

DSNJ117I INITIALIZATION ERROR READING BSDS DSNAME=....., ERROR STATUS=...

Explanation: The log manager encountered an error during initialization reading from the specified bootstrap data set. Error status contains the VSAM return and feedback codes. It is a two byte field with the first containing the hexadecimal return code and the second byte containing the hexadecimal feedback code. Refer to the appropriate VSAM publication for a description of VSAM return and reason codes.

System action: DB2 startup is terminated.

Operator response: Notify the system programmer.

System programmer response: Determine the cause of the read error using the VSAM error status information provided. Restart DB2.

DSNJ118I LOG INITIALIZATION RECEIVED RETURN CODE nnn FROM xxxxxxxx MACRO

Explanation: Log manager initialization received a bad return code from the named macro.

System action: DB2 initialization is terminated.

Operator response: Notify the system programmer.

System programmer response: Determine the problem from the documentation on the named macro and return code. Then take appropriate steps, and restart the DB2 subsystem.

DSNJ119I BOOTSTRAP ACCESS INITIALIZATION PROCESSING FAILED

Explanation: During log manager initialization, the bootstrap data set access function was unable to complete its initialization process. Refer to Part 4 (Volume 1) of *DB2 Administration Guide* for information about recovery from BSDS or log failures.

System action: DB2 startup is terminated.

Operator response: Notify the system programmer.

System programmer response: One or more error messages describing the specific error have preceded this message. Refer to the specific message(s) for error analysis and the appropriate action to take.

DSNJ120I DUAL BSDS DATA SETS HAVE UNEQUAL TIMESTAMPS, BSDS1 SYSTEM=..., UTILITY=..., BSDS2 SYSTEM=..., UTILITY=...

Explanation: When the log manager was initialized, the time stamps of the dual bootstrap data sets (BSDS) did not agree. The time stamps from the system and from the change log inventory (DSNJU003) utility are shown for each BSDS. The time stamps have the format yy.ddd hh:mm:ss.th.

System action: DB2 attempts to re-synchronize the BSDS data sets to restore dual BSDS mode. If re-synchronization is successful, DB2 restart continues.

Operator response: If restart fails, notify the system programmer.

System programmer response: If restart fails, run the print log map (DSNJU004) utility against each BSDS. From the output, determine which data set is obsolete, delete it, define a replacement for it, and copy the remaining BSDS to the replacement.

If output from the print log map (DSNJU004) utility for both data sets is similar, delete the data set with the oldest time stamp, and copy the data set with the most recent time stamp.

DSNJ121I LOG MANAGER ENCOUNTERED AN ERROR READING THE JFCB. DDNAME=*nnnnnnnn*

Explanation: During log manager initialization (if dual BSDS data sets are specified), the job file control block (JFCB) in MVS is read to obtain the data set names associated with DDNAME BSDS1 and BSDS2. This error is caused by a missing DD statement.

System action: DB2 is terminated.

Operator response: Notify the system programmer.

System programmer response: Ensure that a DD statement exists in the DB2 startup procedure for DDNAME BSDS1. If dual BSDS data sets are used, ensure that a DD statement also exists in the DB2 startup procedure for DDNAME BSDS2.

DSNJ122I DUAL BSDS DATA SETS ARE OUT OF SYNCHRONIZATION

Explanation: During log manager initialization, the dual BSDS data sets were found to differ in content.

System action: DB2 startup is terminated.

Operator response: Notify the system programmer.

System programmer response: Run the print log map (DSNJU004) utility against each BSDS to determine which data set was last used as copy 1. Delete the copy 2 data set, define a replacement for the deleted data set, and copy the remaining BSDS to the replacement.

DSNJ123I CHANGE LOG INVENTORY FAILURE DETECTED

Explanation: During log manager initialization, the BSDS data set was found to have been incompletely processed by the change log inventory (DSNJU003) utility.

System action: DB2 startup is terminated.

System programmer response: Run the print log map (DSNJU004) utility to determine what operation against the BSDS did not complete. Run the change log inventory (DSNJU003) utility against the BSDS(s) to allow any unfinished processing to be completed.

DSNJ124I OFFLOAD OF ACTIVE LOG SUSPENDED FROM RBA *xxxxxx* TO RBA *xxxxxx* DUE TO I/O ERROR

Explanation: During off-load, an unrecoverable I/O error was encountered on an active log data set. The data set experiencing the error is marked unusable, and no further logging is done to that data set.

System action: Active log data sets continue to be off-loaded as they become full.

Operator response: Notify the system programmer to allow recovery actions to be initiated.

System programmer response: Recover the data manually from the data set, copy it to an archive data set, run the change log inventory (DSNJU003) utility to make the new archive data set available to the DB2, and remove the error-prone active log data set from DB2 by using the change log inventory utility.

DSNJ125I ERROR DUMPING BSDS, OFFLOAD CONTINUING WITHOUT PRODUCING THE BSDS DUMP

Explanation: An error occurred while dumping the BSDS data set during the off-load process. The data set is not produced, but the off-load process continues. The volume containing the off-loaded data set does not contain a BSDS for recovery use.

System action: The system continues the off-load process without producing a dump of the bootstrap data set.

DSNJ126I BSDS ERROR FORCED SINGLE BSDS MODE

Explanation: The log manager encountered an I/O error or a VSAM logical error on a bootstrap data set. This message is preceded by message DSNJ107I or DSNJ108I.

System action: The log manager enters single BSDS mode using the remaining BSDS.

Operator response: Notify the system programmer.

System programmer response: Execute an offline access method services job to rename the error BSDS and define a new BSDS with the same name. Then enter the log manager -RECOVER BSDS command to reestablish dual BSDS mode.

DSNJ127I SYSTEM TIMESTAMP FOR BSDS=...

Explanation: When the log manager is initialized, the system time stamp for the bootstrap data sets (BSDS) is displayed. The time stamp is of the format yy.ddd hh:mm:ss.th. This time stamp should be close to the last time at which this subsystem was stopped. If not, then it may indicate a restart is being attempted with the wrong BSDS.

System action: DB2 startup continues.

Operator response: If the time displayed does not appear to be close to the time this subsystem was last stopped, contact the system programmer.

System programmer response: If you cannot explain the apparent time discrepancy, cancel the subsystem. From the startup procedure for the subsystem services address space, determine the data set names of the bootstrap data sets and run the print log map (DSNJU004) utility. Check whether the active and archive log data sets all belong to this subsystem. If not, then change the startup procedure for the subsystem services address space to use the correct bootstrap data sets.

DSNJ128I LOG OFFLOAD TASK FAILED FOR ACTIVE LOG *nnnnn*

Explanation: The off-load task abended while attempting to offload the relative byte address (RBA) range in active log data set 'nnnnn'.

System action: The off-load task terminates and the archive data sets allocated to the off-load task are deallocated and deleted. The status of the active log data sets involved in the failed off-load processing remains set to 'not reusable'.

The log off-load task will be reinitiated by one of several events. The most common are:

- All the available space in the current active log data set has been used (normal case).
- A DSNJ110E message is issued.
- The master address space is started, but data in the active log has not been archived.
- An I/O error occurs on the active log, which will force the system to truncate and off-load the active log data set, and switch to a new active log data set.

Operator response: This message is the result of an off-load failure, and will be preceded by one or more DB2 messages (for example, DSNJ073I) and/or MVS ABEND messages (for example, IEC030I, IEC031I, IEC032I). If DB2 is operating with restricted active log resources (see message DSNJ110E), quiesce the system to restrict logging activity until the ABEND or the DSNJ110E condition can be resolved. Notify the system programmer.

System programmer response: This message is the result of an off-load failure, and will be preceded by

one or more DB2 messages (for example, DSNJ073I) and/or MVS ABEND messages (for example, IEC030I, IEC031I IEC032I). Investigate and correct the cause of the ABEND before the off-load is attempted again by DB2.

Problem determination: This message is the result of an off-load ABEND and will be preceded by one or more DB2 messages and/or MVS ABEND messages. See the documentation for the associated DB2 and/or MVS messages to formulate a course of corrective action. Use the print log map (DSNJU004) utility to print the BSDS (both copies if running in dual mode), and then use the DSNJU004 output to determine the current status of the active and archive log data sets.

This message can be generated for a variety of reasons. However, the most likely are:

- Archive log data set allocation errors. Refer to the text for message DSNJ103I for corrective action.
- The size of the archive log data set is too small to contain the active log data set(s) during off-load processing. All secondary space allocations have been used. This condition is normally accompanied by MVS ABEND message IEC030I.
- All available space on the DASD volume(s) to which the archive data set is being written has been exhausted. This condition is normally accompanied by MVS ABEND message IEC032I.
- The primary space allocation for the archive log data set (as specified in ZPARMs) is too large to allocate to any available online DASD device. This condition is normally accompanied by MVS ABEND message IEC032I.

DSNJ129I END OF *log-rba eol-rba* **X COULD NOT BE FOUND IN ANY ACTIVE LOG DATA SET HIGHEST RBA FOUND WAS** *hi-rba* **X**

Explanation: The log manager has been requested to find 'eol-rba', the log record which has been recorded in the bootstrap data set (BSDS) as the highest RBA written. This RBA cannot be found in any active log data set. The highest RBA which could be found in any active data set was 'hi-rba'.

System action: DB2 startup processing is terminated.

Operator response: Notify the system programmer.

System programmer response: Most likely, the active log data set containing the requested RBA has been deleted from the BSDS by the change log inventory (DSNJU003) utility. If the data set has not been reused, run DSNJU003 to add this data set back into the BSDS. Restart DB2.

If the data set is not available, a conditional restart must be performed to either truncate the log prior to the RBA value of the missing data set, or to cold start at a higher RBA value. Refer to Part 4 (Volume 1) of *DB2 Administration Guide* for more information.

DSNJ130I ICF CATALOG NAME IN BSDS DOES NOT AGREE WITH DSNZPARM. BSDS CATALOG NAME=*name***, DSNZPARM CATALOG NAME=***name*

Explanation: The ICF catalog name defined in DSNZPARM does not agree with the name contained in the bootstrap data set (BSDS).

System action: DB2 startup is abended.

User response: Notify the system programmer.

Operator response: If you attempt to start the DB2 subsystem, you will receive this same message and abnormal termination will result until the problem is resolved.

System programmer response: Change the ICF catalog name in the BSDS to agree with the name defined in DSNZPARM during DB2 initialization. To change the ICF catalog name in the BSDS, use the BSDS change log inventory (DSNJU003) utility with the NEWCAT control statement.

Problem determination: During log manager initialization, the ICF catalog name defined in DSNZPARM is compared to the catalog name in the BSDS to ensure that the catalog name used by the log manager during DB2 processing is correct. These names were found not to be the same.

DSNJ131I *csect-name* **DUAL BSDS MODE RESTORED FROM BSDS***n*

Explanation: Dual BSDS mode has been restored using BSDS copy *n*. This copy is the BSDS data set with the most recent system timestamp.

System action: DB2 startup continues.

DSNJ132I *csect-name* **ARCHIVE LOG READ ERROR DSNAME=***data-set-name***,** *diag-information*

Explanation: An I/O error occurred attempting to read from archive log data set *data-set-name*. The value for *diag-information* contains a portion of the message text generated by the SYNADAF macro to describe the error. The generated message text contains information such as jobname, stepname, unit address, device type, ddname, operation attempted, and error description.

System action: DB2 will attempt the read request on a secondary copy of the archive log, if it is available. If a secondary copy of the archive does not exist, or this error occurs on the secondary copy of the archive log, the log read request fails with reason code 00D10329.

Operator response: Notify the system programmer.

System programmer response: If the archive log read request is satisfied by a secondary archive log data set, no immediate response is required. Otherwise, attempt to determine the cause of the I/O error and recover the

archive data set as required. For example, if the archive log data set has been reblocked by moving or copying the data set, or by an HSM recall, it could result in a 'wrong length record' error. Use IEBGENER, or an equivalent program, to restore the data set with the original DCB information.

DSNJ138I *csect-name* **OFFLOAD COMPLETE FOR LOG TRUNCATION FROM AN ARCHIVE LOG COMMAND, ENDRBA=***xxxxxxxxxxxx*

Explanation: This message indicates that the offload of the truncated active logs, resulting from an -ARCHIVE LOG command, has completed successfully. *xxxxxxxxxxxx* is the ENDRBA of the truncated active logs and the newly created archive log data sets.

DSNJ139I LOG OFFLOAD TASK ENDED

Explanation: Processing of the active log off-load ended.

System action: This message is written to the MVS console.

Operator response: This message does not guarantee that the off-load completed without errors. Check the console log and task messages to review the execution of the off-load task to determine whether any abnormal events occurred during the off-load.

DSNJ140I *csect-name* **INVALID BSDS RELEASE LEVEL ACTIVE=***xxxxxx* **ARCHIVE=***yyyyyy* **DSN=***zzzzzz*

Explanation: At initialization time DB2 detected that the BSDS log records are not at the proper release level for this release of DB2. 'xxxxxx' contains the release level for the ACTIVE log records and 'yyyyyy' contains the release level for the ARCHIVE log records. 'zzzzzz' contains the data set name for the BSDS encountering the error.

System action: DB2 startup will be terminated.

Operator response: Notify the system programmer so operating procedures can be changed.

System programmer response: Run the appropriate conversion program (DSNJMIG0 or DSNJFAL0) to convert the BSDS data set to a compatible release level.

DSNJ150E LOG CAPTURE EXIT ABEND, EXIT DEACTIVATED

Explanation: An abnormal program interrupt was detected while executing in the installation-supplied log capture exit code (i.e. entry point DSNJW117 in load module DSNJL004). As a result of the abend, the log capture exit will no longer be active; log data will no longer be available for Exit capture/processing. This message can only occur when an installation-supplied log capture exit (entry DSNJW117) is active for this instance of DB2.

System action: The log capture exit (entry point DSNJW117) is terminated. No further calls will be attempted for this DB2 instance. A full dump is provided for diagnostic purposes.

User response: See "System Programmer Response" below.

Operator response: Notify the system programmer.

System programmer response: Determine the cause of the DSNJL004 load module (DSNJW117 entry point) abend and take corrective action.

Attention: A correctly-functioning copy of load module DSNJL004/entry DSNJW117 must be available in order to start DB2. If the problem which caused the above abend cannot be corrected, ensure that the default DSNJW117 entry (load module DSNJL004 - supplied with the DB2 release) is available during the next DB2 subsystem start.

DSNJ151I *csect-name* **IN READING** *log-type log-value*, **A NONZERO REASON CODE WAS RETURNED. CONNECTION-ID=***xxxx*, **CORRELATION-ID=***yyyyyy*, **REASON-CODE=***ccc*, **MEMBER-ID=***member-id*

Explanation: DB2 could not successfully complete the read of this *log-value* and *log-type* (RBA or LRSN) due to reason code 'ccc'. The *member-id* contains the ID of the data sharing member whose logs were being requested at the time of the failure or zero if data sharing is not active. The user or application that encountered the error is identified by the connection and correlation ID. Messages that have the same connection ID and correlation ID relate to the same application. Correlation IDs beginning with '0nn', where 'nn' is a number from 01 to 28, identify system agents.

System action: DB2 attempts to recover from the error.

System programmer response: If DB2 was able to recover from the error and successfully complete the application, no further action is required. If the application abnormally terminated or DB2 could not recover successfully, this message is followed by one or more messages. Refer to the information in this message and the subsequent messages to determine the appropriate corrective action. For information about recovery from log failures, refer to Part 4 (Volume 1) of *DB2 Administration Guide* .

DSNJ152I *csect-name* **ERROR ENCOUNTERED WHEN ATTEMPTING TO BUILD ARCHIVE LOG VOLUME LIST. CONNECTION-ID=***xxxx*, **CORRELATION-ID=***yyyyyy*, **REASON-CODE=***ccc*

Explanation: An error occurred while attempting to create the Archive Log Volumes report. An RBA range could not be successfully mapped into one or more archive data sets due to reason code 'ccc'. The user or application that encountered the error is identified by the connection and correlation ID. This message might be preceded by one or more related error messages. Messages that have the same connection ID and correlation ID relate to the same user. Correlation IDs beginning with '0nn', where 'nn' is a number from 01 to 28, identify system agents.

This failure could be caused by one or more missing archive log data sets, or a system error (for example, an I/O error reading BSDS).

The Archive Log Volumes report (see message DSNJ330) is not produced. In addition, no premounting of tapes is possible.

System action: The user or application continues execution. The physical read process for the user or application continues until the job completes normally or terminates abnormally. The job can terminate abnormally if the error is encountered again when the data set is physically required for the read process.

System programmer response: If the user or application completes successfully, no further action is necessary. If the user or application does not complete successfully, refer to the messages related to the actual failure to determine the appropriate corrective actions. For information about recovery from log failures, refer to Part 4 (Volume 1) of *DB2 Administration Guide*.

DSNJ153E *csect-name* **CRITICAL LOG READ ERROR CONNECTION-ID =** *connection-id* **CORRELATION-ID =** *correlation-id* **LUWID =** *logical-unit-of-work-id*=*token* **REASON-CODE =** *reason-code*

Explanation: While processing a DB2 must-complete function, DB2 encountered an error during an attempt to access the required log data sets.

CONNECTION-ID, CORRELATION-ID, and LUWID together identify the thread associated with the request. *token* is a unique token number associated with the LUWID. *reason-code* identifies the most recent log-read failure.

System action: The DSNJ154I message requests to retry the log-read request, and immediately follows this error message. DB2 will wait for a reply to message DSNJ154I before retrying the failed log-read request, or before abending.

System programmer response: Review the description of the reason-code associated with the log-read failure. Examine the system console log for additional messages regarding the log- read error. Correct the cause of the error, if possible, and reply 'Y' to the DSNJ154I message in order to retry the log-read request. If it is not possible to correct the problem causing the log-read errors, and the failing log read request is the result of a thread rollback or -RECOVER POSTPONED command, consider canceling the log read request with the -CANCEL THREAD NOBACKOUT or -RECOVER POSTPONED CANCEL command. Please see the command reference manual for details about these commands. If canceling the log read request is not an option, quiesce the work on the DB2 system before replying 'N' to the DSNJ154I message in preparation for DB2 termination.

DSNJ154I *csect-name* **REPLY Y TO RETRY LOG READ REQUEST, N TO ABEND**

Explanation: While processing a DB2 must-complete function, DB2 encountered an error during an attempt to access the required log data sets.

System action: The DSNJ153E message identifies the critical log-read error, and immediately precedes this message. DB2 will wait for the operator's reply to proceed with the log-read request.

Operator response: Notify the system programmer. Once actions have been taken to correct the cause of the log-read errors, reply 'Y' to request a retry of the failing log-read request. A reply of 'N' will fail the log-read request and DB2 will abend.

System programmer response: Review the description of the reason-code associated with the log-read failure, and examine the system console log for additional messages regarding the log-read error. Correct the cause of the error, if possible, and reply 'Y' to the DSNJ154I message to retry the log-read request. If it is not possible to correct the problem causing the log-read errors, quiesce the work on the DB2 system before replying 'N' to the DSNJ154I message in preparation for DB2 termination.

DSNJ155I MAXARCH SUBSYSTEM PARAMETER VALUE OF nnnn IS TOO LARGE

Explanation: The MAXARCH subsystem parameter that was specified is too large for the current BSDS format. MAXARCH, which is specified as RECORDING MAX on panel DSNTIPA, indicates the maximum number of archive log volumes per log copy to record in the BSDS. The maximum value depends on whether or not you have performed a BSDS conversion. The limit for archive volumes recorded in the BSDS is 1,000 per copy if no conversion has occurred, or 10,000 per copy if the BSDS has been converted.

System action: The MAXARCH subsystem parameter is set to 1,000 archive log volumes, and restart continues.

Operator response: Notify the system programmer.

DSNJ156I INVALID BSDS CONVERSION, DSN=*xxxxxx*

Explanation: During initialization time, DB2 detected an invalid BSDS. An invalid BSDS can result from a failure during a prior attempt to run conversion program DSNJCNVB. *xxxxxx* is the data set name for the BSDS encountering the error.

System action: DB2 startup will be terminated.

System programmer response: The procedure for running program DSNJCNVB includes making a copy of the BSDS prior to conversion. Restore the BSDS to the original pre-conversion copy, and retry the conversion. Once the conversion is successful, restart DB2.

DSNJ200I *csect-name xxxxxxxx* **UTILITY PROCESSING COMPLETED SUCCESSFULLY**

Explanation: The control section 'csect-name' successfully completed the 'xxxxxxxx' utility.

System action: DB2 writes this message to the SYSPRINT data set, not to a console.

DSNJ201I *csect-name xxxxxxxx* **UTILITY PROCESSING WAS UNSUCCESSFUL**

Explanation: The control section 'csect-name' was unable to successfully complete processing of 'xxxxxxxx' utility.

System action: The current utility is terminated. DB2 writes this message to the SYSPRINT data set, not to a console.

User response: Review other messages produced by the utility to determine the appropriate action to be taken.

DSNJ202I *csect-name* **INSUFFICIENT VIRTUAL STORAGE AVAILABLE TO CONTINUE WITH UTILITY**

Explanation: A GETMAIN macro instruction has failed, because virtual storage is unavailable for the job-step.

System action: The current utility is terminated. This message is written to the SYSPRINT data set, not to a console.

User response: Rerun the utility after providing a larger virtual storage allocation for the job-step.

DSNJ203I xxxxxxxx OPERATION IS INVALID

Explanation: The user entered a utility control statement operation of 'xxxxxxxx' that is invalid.

System action: The current utility is terminated. This message is written to the SYSPRINT data set, not to a console.

User response: Correct the control statement, and rerun the utility.

DSNJ204I xxxxxxxx PARAMETER IS INVALID

Explanation: The user specified a utility control statement parameter of 'xxxxxxxx' that is invalid.

System action: The current utility is terminated. This message is written to the SYSPRINT data set, not to a console.

User response: Correct the control statement, and rerun the utility.

DSNJ205I xxxxxxxx PARAMETER HAS NO ARGUMENT

Explanation: 'xxxxxxxx' contains the name of a parameter that requires an argument.

System action: The current utility is terminated. This message is written to the SYSPRINT data set, not to a console.

User response: Specify an argument for the identified parameter and then rerun the utility.

DSNJ206I xxxxxxxx PARAMETER REQUIRES NO ARGUMENT

Explanation: 'xxxxxxxx' contains the name of the parameter that has been incorrectly followed by an = symbol.

System action: The current utility is terminated. This message is written to the SYSPRINT data set, not to a console.

User response: Correct the control statement, and rerun the utility.

DSNJ207I *csect-name* **PARAMETERS INCONSISTENT WITH SPECIFIED OPERATION**

Explanation: The user has specified utility control statement parameters that are inconsistent with the specified utility operation.

System action: The current utility is terminated. This message is written to the SYSPRINT data set, not to a console.

User response: Correct the control statement, and rerun the utility.

DSNJ208E *csect name* **MEMBER** *member name* **WAS NOT FOUND IN THE BSDS SPECIFIED IN THE GROUP DD STATEMENT.**

Explanation: You are running the print log map (DSNJU004) utility. No member record exists for this member in the bootstrap data set (BSDS) specified in the GROUP DD statement.

System action: The log map for this member is not printed. If the records for other members are requested and are found in the BSDS specified in the GROUP DD statement, they are printed.

User response: Use the DISPLAY GROUP command to verify that the member name is a valid name and that the member has successfully joined the group. Also verify that the BSDS specified in the GROUP DD statement is correct. If the member name and the BSDS name are correct, then the most likely causes of this message are:

- The 'missing member' never joined the group because it was never started.
- The member whose BSDS you specified in the GROUP DD statement was down when the missing member joined the group and has not been restarted.

To gather the information from the missing member, you can rerun the utility, using the BSDS DDNAME option to point to the missing member's BSDS. To avoid this problem in the future, make sure the BSDS named in the GROUP statement is from a recently started member.

DSNJ209I ENDRBA VALUE MUST BE GREATER THAN STARTRBA VALUE

Explanation: The 'STARTRBA' and 'ENDRBA' parameters specify an RBA range. Hence, the ENDRBA value must be greater than the STARTRBA value.

System action: The current utility is terminated. This message is written to the SYSPRINT data set, not to a console.

User response: Correct the control statement, and rerun the utility.

DSNJ210I OLD ICF CATALOG NAME=oldname, NEW CATALOG NAME=newname

Explanation: Change log inventory (DSNJU003) utility processing for the NEWCAT control statement has completed successfully.

oldname
Either:

- The name of the prior ICF catalog or
- 'NULL' if no prior name existed.

newname
The name of the ICF catalog defined by the VSAMCAT keyword.

DSNJ211I UNEXPECTED END OF DATA ON SYSIN DATA SET

Explanation: Additional control statements were expected but could not be found.

System action: The current utility is terminated. This message is written to the SYSPRINT data set, not to a console.

User response: Correct the control statement(s), and rerun the utility.

DSNJ212I *csect-name* **ERROR RETURNED FROM BSDS READ, RPLERRCD=***yy***, DDNAME=***xxxxxxxx*

Explanation: A VSAM GET was issued that resulted in a nonzero return code. 'yy' contains the error code returned by VSAM. 'xxxxxxxx' contains the ddname of the BSDS encountering the error.

System action: The current utility is terminated. This message is written to the SYSPRINT data set, not to a console.

User response: The action taken is dictated by the return code. The BSDS may have to be recovered by use of a backup copy.

DSNJ213I *csect-name* **ERROR RETURNED FROM BSDS WRITE, RPLERRCD=***yy***, DDNAME=***xxxxxxxx*

Explanation: A VSAM PUT was issued that resulted in a nonzero return code. 'yy' contains the error code returned by VSAM. 'xxxxxxxx' contains the ddname of the BSDS encountering the error.

System action: The current utility is terminated. This message is written to the SYSPRINT data set, not to a console.

User response: The action to be taken is dictated by the return code. The BSDS may have to be recovered by use of a backup copy.

DSNJ214I SPECIFIED DSNAME ALREADY EXISTS IN BSDS, DDNAME=xxxxxxxx

Explanation: The user attempted a NEWLOG operation with a data set name that already exists in the BSDS. An entry is never made in a BSDS if the specified DSNAME currently exists in either the active or archive records of that BSDS. 'xxxxxxxx' contains the ddname of the subject BSDS.

System action: The current utility is terminated. This message is written to the SYSPRINT data set, not to a console.

User response: Either correct the control statement and rerun the utility, or delete the existing DSNAME from the BSDS and rerun the utility.

DSNJ215I *csect-name* **SVC 99 INFORMATION REQUEST FOR DD NAME '***ddname***' FAILED. RC=** *rc*, **RSN=** *rsncd*, **AND INFO CODE** *infocode*.

Explanation: The Print Log Map utility requested information from the MVS Dynamic Allocation facility about the DD name shown in the message, but the system was unable to provide the information. The return code, reason code, and information code were returned by the system's SVC 99 facility. See *MVS/ESA Programming: Authorized Assembler Services Guide* for an explanation of these codes.

System action: If the information request was for the SYSIN DD statement, the Print Log Map utility assumes the control statement was not provided and attempts to continue the request using the SYSUT1 DD statement to locate the BSDS to be mapped.

If the information request was for the DD name of a BSDS, attempts to get information about BSDS data sets is terminated. If information about any BSDS data sets was previously acquired, the maps of those BSDS data sets are printed. Otherwise, the utility terminates.

System programmer response: Use the MVS manual (or the equivalent manual for the level of MVS on your system) to help you in problem determination.

Problem determination: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659 : 4, 32.

DSNJ216I NO SPACE IN BSDS FOR ADDITIONAL ACTIVE LOG ENTRIES, DDNAME=*xxxxxxxx*

Explanation: The maximum number of active log data sets is fixed. No further entries can be inserted after the maximum has been reached. *xxxxxxxx* contains the ddname of the subject BSDS.

System action: The current utility is terminated. This message is written to the SYSPRINT data set, not to a console.

User response: Run the print log map utility to determine the current status of the BSDS. Subsequent actions can then be formulated, depending upon the status of the BSDS.

DSNJ217I SPECIFIED DSNAME DOES NOT EXIST IN BSDS, DDNAME=xxxxxxxx

Explanation: The DELETE operation specifies a DSNAME that cannot be found in the BSDS. 'xxxxxxxx' contains the ddname of the subject BSDS.

System action: The current utility is terminated. This message is written to the SYSPRINT data set, not to a console.

User response: Correct the control statement, and rerun the utility.

DSNJ218I SPECIFIED VOLUME DOES NOT EXIST IN BSDS, DDNAME=xxxxxxxx

Explanation: The DELETE operation specifies a COPY1VOL or COPY2VOL argument that cannot be found in the BSDS. 'xxxxxxxx' contains the ddname of the subject BSDS.

System action: The current utility is terminated. This message is written to the SYSPRINT data set, not to a console.

User response: Correct the control statement, and rerun the utility.

DSNJ219I *csect-name* **OPEN ERROR, DDNAME=***xxxxxxxx*

Explanation: An error occurred when the CSECT 'csect-name' tried to open a data set named 'xxxxxxxx'.

This error can be caused by a number of different conditions. The most probable conditions are:

1. The ddname of the SYSPRINT, SYSIN, or SYSUT1 data set was not specified in the user's job control language (JCL).
2. The DB2 subsystem is currently executing.
3. The bootstrap data set (BSDS) was allocated by another job with a disposition (DISP) that conflicts with the DISP specified in the user's JCL.
4. The data set associated with 'xxxxxxxx' is already open, possibly due to an earlier system abend.
5. The user is not authorized to access the data set associated with 'xxxxxxxx'
6. Insufficient virtual storage is available to perform the OPEN command.
7. The catalog indicates that the data set associated with 'xxxxxxxx' has an invalid physical record size.

System action: The current utility is terminated.

DB2 writes this message to the SYSPRINT data set, not to a console.

User response: The user's action depends on the condition that caused the OPEN error. The following is a list of appropriate actions corresponding to the conditions listed in the explanation:

1. Provide the missing DD statements, and then rerun the utility. Refer to *DB2 Utility Guide and Reference* for details about the required DD statements.
2. Wait until the DB2 subsystem is inactive before running the utility again because the log utility cannot execute while the DB2 subsystem is active.
3. Correct the disposition conflict and then rerun the utility.
4. Submit an access method services (IDCAMS) VERIFY job against the data set associated with 'xxxxxxxx'. Rerun the log utility job.
5. In the case of an authorization failure, a separate message is usually generated from the authorization facility (RACF, for example). Investigate the authorization messages and obtain the proper authorization before running the utility again.
6. Insufficient virtual storage is usually accompanied by a separate error from the MVS operating system. Increase the REGION size on the JCL job step and rerun the utility.
7. Contact the system programmer.

DSNJ220I BSDS IN CREATE MODE. NO DATA TO MAP, DDNAME=*nnnnnnnn*

Explanation: A utility found the BSDS to be in create mode, meaning it contains no data. *nnnnnnnn* contains the ddname of the BSDS data set.

System action: The current utility is terminated. This message is written to the SYSPRINT data set, not to a console.

User response: Ensure that the BSDS data set name specified as input to the utility is correct.

DSNJ221I PREVIOUS ERROR CAUSED xxxxxxxx OPERATION TO BE BYPASSED

Explanation: DB2 encountered errors while processing a utility. These errors subsequently caused 'xxxxxxxx' to be bypassed.

This message is a warning only and is displayed after messages that specify the error or errors that caused the utility operation to abend. Note that the error or errors may not be associated with the current 'xxxxxxxx' operation; rather, under log utility processing, a significant error in any operation causes the control statements for this and any subsequent operations to be checked for syntax only. BSDS updates do not occur for any operation specified in this message.

System action: The log utility continues to process. However, for this and all subsequent operations, the bootstrap data set (BSDS) is not updated and the utility only checks the syntax of the control statements.

DB2 writes this message to the SYSPRINT data set, not to a console.

User response: Consult the previous messages and correct any errors that caused this message to be generated. Resubmit the log utility job for all operations that have been bypassed.

DSNJ222I INVALID SPECIFICATION OF xxxx PARAMETER ARGUMENT

Explanation: You specified the parameter 'xxxx'. This parameter is not valid for the argument.

System action: The current utility is terminated. This message is written to the SYSPRINT data set, not to a console.

User response: Correct the parameter argument on the control statement, and rerun the utility.

DSNJ223I xxxx PARAMETER ARGUMENT EXCEEDS MAXIMUM ALLOWABLE LENGTH

Explanation: The argument value of the parameter named 'xxxx' exceeds the maximum length allowed.

System action: The current utility is terminated. This message is written to the SYSPRINT data set, not to a console.

User response: Correct the parameter argument on the control statement, and rerun the utility.

DSNJ224I xxxx PARAMETER APPEARS TOO OFTEN

Explanation: The parameter named 'xxxx' was specified more than once on the same control statement.

System action: The current utility is terminated. This message is written to the SYSPRINT data set, not to a console.

User response: Remove the redundant parameter, and rerun the utility.

DSNJ225I xxxx OPERATION COMPLETED SUCCESSFULLY

Explanation: The change log inventory (DSNJU003) utility operation named 'xxxx' completed successfully.

System action: This message is written to the SYSPRINT data set, not to a console.

DSNJ226I SPECIFIED VOLUME ALREADY EXISTS IN BSDS, DDNAME=xxxxxxx

Explanation: The specified volume currently exists in the archive log records of the BSDS. 'xxxxxxxx' specifies the ddname of the subject BSDS.

System action: The current utility is terminated. This

message is written to the SYSPRINT data set, not to a console.

User response: Either correct the parameter argument on the control statement, or delete the specified volume and rerun the utility.

DSNJ227I NO SPACE AVAILABLE IN BSDS FOR ADDITIONAL ARCHIVE VOLUME ENTRIES, DDNAME=xxxxxxxx

Explanation: The maximum number of archive volumes has been exceeded, and no more space is available for volume entries in the copy specified.

System action: The current utility is terminated.

User response: Delete some of the archive entries in the specified copy number, and rerun the utility.

DSNJ228I *csect-name* **LOG DEALLOCATION ERROR DSNAME=***dsname***, ERROR STATUS=***eeeeiiii***, SMS REASON CODE=***ssssssss***

Explanation: The log manager encountered an error while trying to dynamically deallocate the data set. Error status is the error reason code returned by MVS dynamic allocation.

System action: Log manager processing continues.

Operator response: Notify the system programmer.

System programmer response: The error status portion of this message contains the 2-byte error code (S99ERROR) followed by the 2-byte information code (S99INFO) from the SVC request block. If the S99ERROR code indicates an SMS deallocation error (97XX), then 'ssssssss' contains additional SMS reason code information obtained from S99ERSN. Refer to the *MVS/ESA SPL: Application Development Guide* for a description of these codes.

DSNJ229I *csect-name* **OPEN FAILED FOR SYSIN**

Explanation: An error occurred when the DB2 Print Log Map utility tried to open the SYSIN data set.

Possible causes are:

1. The DD name of the SYSIN data set was not specified in the user's job control language (JCL).
2. Insufficient virtual storage is available to perform the OPEN command.
3. The characteristics of the SYSIN data set are different from those expected by the print log map (DSNJU004) utility.

System action: DSNJU004 attempts to continue processing without an input control statement. DB2 writes this message to the SYSPRINT data set, not to the system console.

User response: Take one of the following actions based on the cause of the error.

1. Provide the missing DD statement and rerun the utility. Refer to *DB2 Utility Guide and Reference* for details about the required DD statements.
2. Insufficient virtual storage is usually accompanied by a separate error message from the MVS operating system. Increase the REGION size on the JCL job step and rerun the utility.
3. Check that the SYSIN data set is a fixed record format (blocked or unblocked) with an LRECL of 80. You might need to contact the system programmer for help if this is a JES data set.

DSNJ230I LOG OFFLOAD SUBCOMPONENT INITIALIZATION PROCESSING FAILED

Explanation: During log manager initialization, the off-load function was unable to complete its initialization process.

System action: DB2 startup is terminated.

Operator response: Notify the system programmer.

System programmer response: One or more error messages describing the specific error preceded this message. Refer to the specific message(s) for error analysis and the appropriate actions to take.

DSNJ231I LOG COMMAND INITIALIZATION PROCESSING FAILED

Explanation: During log manager initialization, the log command function was unable to complete its initialization process.

System action: DB2 startup is terminated.

Operator response: The rest of subsystem termination will direct the actions you should take.

System programmer response: One or more error messages describing the specific error preceded this message. Refer to the specific message(s) for error analysis and the appropriate action to take.

This message will be of interest to the IBM support center when the problem is described.

DSNJ232I OUTPUT DATA SET CONTROL INITIALIZATION PROCESSING FAILED

Explanation: During log manager initialization, the output data set control function was unable to complete its initialization process.

System action: DB2 startup is terminated.

System programmer response: One or more error messages describing the specific error preceded this message. Refer to the specific message for error analysis and the appropriate action to take.

This message will be of interest to the IBM support center when the problem is described.

DSNJ233I ARCHIVE LOG READ INITIALIZATION PROCESSING FAILED

Explanation: During log manager initialization, the archive log read function was unable to complete its initialization process.

System action: DB2 startup is terminated.

Operator response: Notify the system programmer.

System programmer response: One or more error messages describing the specific error preceded this message. Refer to the specific message(s) for error analysis and the appropriate action to take.

DSNJ234I ARCHIVE LOG COMMAND QUIESCE INITIALIZATION PROCESSING FAILED

Explanation: During log manager initialization, the quiesce function which supports the -ARCHIVE LOG MODE(QUIESCE) command processing was unable to complete its initialization process.

System action: DB2 startup is terminated.

Operator response: Notify the system programmer.

System programmer response: One or more error messages describing the specific error preceded this message. Refer to the specific message(s) for error analysis and the appropriate action to take.

DSNJ235I OUTPUT BUFFER WRITER INITIALIZATION PROCESSING FAILED

Explanation: During log manager initialization, the output buffer writer function was unable to complete its initialization process.

System action: DB2 startup is terminated.

Operator response: Notify the system programmer.

System programmer response: One or more error messages describing the specific error preceded this message. Refer to the specific message(s) for error analysis and the appropriate action to take.

DSNJ236I BOOTSTRAP ACCESS TERMINATION PROCESSING FAILED

Explanation: During log manager termination, the bootstrap data set access function was unable to complete its termination process.

System action: Log manager shutdown continues to the next step.

Operator response: Notify the system programmer.

System programmer response: One or more error messages describing the specific error preceded this message. Refer to the specific message(s) for error analysis and the appropriate action to take.

DSNJ237I *csect-name* **INVALID CONTROL STATEMENT. THE FIRST WORD MUST BE 'MEMBER'.**

Explanation: The print log map (DSNJU004) utility read an invalid control statement from the SYSIN data set.

System action: DSNJU004 ignores the control statement and attempts to continue processing by looking for a SYSUT1 DD statement to describe the BSDS that is to be printed.

User response: Refer to *DB2 Utility Guide and Reference* for information on creating a print log map (DSNJU004) utility control statement. Correct or remove the control statement and rerun the job.

Problem determination: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659 : 32.

DSNJ238I LOG OFFLOAD TERMINATION PROCESSING FAILED

Explanation: During log manager termination, the off-load function was unable to complete its termination process.

System action: Log manager shutdown continues to the next step.

Operator response: Notify the system programmer.

System programmer response: One or more error messages describing the specific error preceded this message. Refer to the specific message(s) for error analysis and the appropriate action to take.

DSNJ239I LOG COMMAND TERMINATION PROCESSING FAILED

Explanation: During log manager termination, the log command function was unable to complete its termination process.

System action: Log manager shutdown continues to the next step.

Operator response: Notify the system programmer.

System programmer response: One or more error messages describing the specific error preceded this message. Refer to the specific message(s) for error analysis and the appropriate action to take.

DSNJ240I OUTPUT DATA SET CONTROL TERMINATION PROCESSING FAILED

Explanation: During log manager termination, the output data set control function was unable to complete its termination process.

System action: The log manager shutdown continues to the next step.

Operator response: Notify the system programmer.

System programmer response: One or more error messages describing the specific error preceded this message. Refer to the specific message(s) for error analysis and the appropriate action to take.

DSNJ241I ARCHIVE LOG READ TERMINATION PROCESSING FAILED

Explanation: During log manager termination, the archive log read function was unable to complete its termination process.

System action: Log manager shutdown continues to the next step.

Operator response: Notify the system programmer.

System programmer response: One or more error messages describing the specific error preceded this message. Refer to the specific message(s) for error analysis and the appropriate action to take.

DSNJ242I ARCHIVE LOG COMMAND QUIESCE TERMINATION PROCESSING FAILED

Explanation: During log manager termination, the quiesce function which supports the -ARCHIVE LOG MODE(QUIESCE) command processing was unable to complete its termination process.

System action: Log manager shutdown continues to the next step.

Operator response: Notify the system programmer.

System programmer response: One or more error messages describing the specific error preceded this message. Refer to the specific message(s) for error analysis and the appropriate action to take.

DSNJ243I OUTPUT BUFFER WRITER TERMINATION PROCESSING FAILED

Explanation: During log manager termination, the output buffer writer function was unable to complete its termination process.

System action: Log manager shutdown continues to the next step.

Operator response: Notify the system programmer.

System programmer response: One or more error messages describing the specific error preceded this message. Refer to the specific message(s) for error analysis and the appropriate action to take.

DSNJ244I LOG TERMINATION RECEIVED RETURN CODE nnn FROM xxxxxxxx MACRO

Explanation: Log manager termination was passed a bad return code from the named macro.

System action: Log manager termination processing continues.

Operator response: Record the return code and macro-name, and notify the system programmer.

System programmer response: This message is associated with a failure in the named macro. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNJ245I CONDITIONAL RESTART RECORD INDICATES TRUNCATION AT *xxx rrr*. REPLY *Y* TOCONTINUE, *N* TO CANCEL

Explanation: The conditional restart control record in use indicates that the DB2 log should be truncated at the specified RBA or LRSN.*xxx* is replaced by either RBA or LRSN, and *rrr* is the value specified in the current record.

System action: If *Y*, DB2 startup continues. If *N*, DB2 startup is terminated.

Operator response: Reply *N* if the truncation is going to occur at an undesirable point. Notify system programmer. Reply *Y* to continue the restart.

System programmer response: Rerun the change log inventory (DSNJU003) utility to modify the conditional restart record.

DSNJ246I CONDITIONAL RESTART RECORD INDICATES COLD START AT RBA rrr. REPLY Y TO CONTINUE, N TO CANCEL

Explanation: The conditional restart control record in use indicates that DB2 is to be cold started and that logging is to be begin at the specified RBA.

System action: If 'Y', DB2 startup continues. If 'N', DB2 startup is terminated.

Operator response: Reply 'N' if the truncation is going to occur at an undesirable point. Notify the system programmer. Reply 'Y' to continue the cold start.

System programmer response: Rerun the change log inventory (DSNJU003) utility to modify the conditional restart record.

DSNJ247E *csect-name* **I/O ERROR DURING PROCESSING OF** *rrr* **RC =** *rc*, **REASON =** *reason*

Explanation: The I/O error occurred at record 'rrr'. 'rc' indicates the return code received from the I/O operation. 'reason' indicates the reason code received from the I/O operation.

If a return code '4' was received, this indicates that the log manager detected a problem. If a return code '8' was received, this indicates a VSAM error.

System action: DB2 startup terminates.

Operator response: Notify the system programmer.

System programmer response: For a return code of '4', if you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem. For a return code of '8', execute an offline access method services job to determine the cause of the VSAM error.

DSNJ248I **ARCHIVE LOG READ DEALLOCATION INITIALIZATION PROCESSING FAILED**

Explanation: During log initialization, the archive log read deallocation process was unable to complete its initialization.

System action: DB2 startup is terminated.

Operator response: Notify the system programmer.

System programmer response: One or more error messages describing the specific error precede this message. Refer to these messages for error analysis and the appropriate action to take.

DSNJ249I **ARCHIVE LOG READ SCHEDULER INITIALIZATION PROCESSING FAILED**

Explanation: During log initialization, the archive log read scheduling process was unable to complete its initialization.

System action: DB2 startup is terminated.

Operator response: Notify the system programmer.

System programmer response: One or more error messages describing the specific error precede this message. Refer to these messages for error analysis and the appropriate action to take.

DSNJ250E *csect-name* **DATA SHARING REQUIRES ALL ACTIVE LOG DATA SETS TO BE DEFINED WITH SHAREOPTIONS(2 3) OR HIGHER.**

Explanation: DB2 detected an active log data set defined with share options that do not permit recovery operations in a data sharing environment. This message is followed by message DSNJ251E, which identifies the specific active log data set.

All active log data sets for a DB2 data sharing member must be defined with a minimum of SHAREOPTIONS(2 3) for successful DB2 initialization. See the description of message DSNJ251E for more details.

System action: The DB2 subsystem detecting this condition is terminated.

DSNJ251E *csect-name* **DATA SET** *dsname* **FAILS TO MEET THE MINIMUM REQUIRED SHAREOPTIONS FOR DATA SHARING.**

Explanation: The share options defined for the data set named in the message are too low to permit the data set to participate in a DB2 data sharing group. All active log data sets must be defined with SHAREOPTIONS(2 3) or higher in order to participate in a data sharing group.

System action: The DB2 subsystem detecting this error is terminated.

System programmer response: Using IDCAMS LISTCAT or TSO LISTCAT, display the share options of all of the active log data sets of all of the members in the data sharing group. For any active log data set that is defined with less than SHAREOPTIONS(2 3), use IDCAMS ALTER or TSO ALTER to alter the data set definitions to SHAREOPTIONS(2 3) or higher.

Problem determination: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Collect the following diagnostic items listed in Appendix A, “Problem determination,” on page 659: 1, 2, 5, 11.

DSNJ253I *csect-name alias-port* **USE IS INVALID**

Explanation: The *alias-port* value has been used in an invalid manner. The error could be due to one of the following conditions:

- *alias-port* was specified after more than one alias-name parameter
- *alias-port* duplicates the port value for the PORT or RESPORT parameters which have been specified with this update, or are already stored in the BSDS

System action: The update of the BSDS will not be performed.

User response: Retry the BSDS update with unique *alias-port* values, which also do not match the values for the PORT or RESPORT parameters.

DSNJ254I *xxxx* **NUMBER OF PARAMETER ARGUMENTS EXCEED MAXIMUM VALUE**

Explanation: The number of arguments for the parameter named *xxxx* exceed the maximum value allowed.

System action: The current operation is terminated. This message is written to the SYSPRINT data set, but not written to a console.

User response: Change the number of parameter arguments on the control statement, and rerun the utility.

DSNJ260I **csect_name BSDS CONVERSION FOR DDNAME=***nnnnnnnn* **COMPLETED SUCCESSFULLY**

Explanation: The BSDS conversion utility issues this message to indicate a successful conversion for the BSDS data set specified by the *nnnnnnnn* DDNAME.

System action: Processing continues.

DSNJ261I **csect_name BSDS CONVERSION FOR DDNAME=***nnnnnnnn* **WAS NOT SUCCESSFUL**

Explanation: The BSDS conversion utility issues this message to indicate conversion was not successful for the BSDS data set specified by the *nnnnnnnn* DDNAME.

System action: The action taken depends on whether dual BSDS data sets are specified, and if so, which conversion failed. The DDNAME in the message indicates the failing BSDS data set.

If dual BSDS data sets are specified, the SYSUT1 data set is converted first. If that conversion is successful, the SYSUT2 data set is then converted. A failure at this point does not affect the successful SYSUT1 conversion.

If conversion fails for the SYSUT1 data set, processing is terminated at that point. In this case, any SYSUT2 data set that was specified will not be converted.

User response: Review other messages produced by the utility to determine the cause of the failure and the resulting action to take.

DSNJ266I **WARNING - BSDS UTILITY TIMESTAMP MISMATCH DETECTED. PROCESSING TERMINATED.**

Explanation: DB2 detected a mismatch in the timestamps for the SYSUT1 and SYSUT2 BSDS copies during execution of BSDS conversion utility DSNJCNVB. This indicates the possibility that the dual BSDSs are out of sync.

System action: The program will terminate with no action taken.

System programmer response: Run the print log map utility (DSNJU004) against each BSDS. From the output, determine which data set is obsolete, delete it, and define a replacement for it. Then copy the remaining data set into the replacement and retry the utility.

If output from the print log map utility for both data sets is similar, delete the data set with the oldest time stamp, and copy the data set with the most recent time stamp into the replacement.

DSNJ301I *csect-name* **ERROR** *action* **ONLINE BOOTSTRAP DATA SET**

Explanation: During command processing for the -RECOVER BSDS command or the -ARCHIVE LOG command, an error occurred while performing an operation on the Bootstrap Data Set. The type of operation is specified in the 'action' keyword in the message text:

- **ACCESSING** Unable to OPEN the BSDS.
- **READING** Unable to read a required record from the BSDS.
- **UPDATING** Unable to write a required record to the BSDS.
- **RESTORING DUAL** The contents of the stable BSDS was successfully copied to the replacement BSDS; however, the DB2 subsystem was unable to successfully restore dual BSDS operation.

System action: If this message was received during processing of the -RECOVER BSDS command, then the DB2 subsystem will continue in single BSDS mode. If this message was received during processing of the -ARCHIVE LOG command, the archive log history record in the BSDS will not be updated to reflect the occurrence of an -ARCHIVE LOG command; logging and the off-load processing will continue.

Operator response: Inform the system programmer of the type of error that occurred, and whether the error occurred while processing a -RECOVER BSDS command or an -ARCHIVE LOG command.

System programmer response: If this message was received during processing of the -RECOVER BSDS command, then recovery action must be performed on the bootstrap data set before reentering the command. If this message was received during processing of the -ARCHIVE LOG command, no action is necessary.

DSNJ302I **ALLOCATION ERROR ON REPLACEMENT BSDS DSNAME=... ERROR STATUS=...**

Explanation: The RECOVER BSDS command encountered an error while trying to dynamically allocate the specified data set. DSNAME is the data set name. Error Status is the error code and information

code returned by MVS dynamic allocation.

System action: Processing of the command is terminated. The system continues in single BSDS mode.

Operator response: Determine the cause of the error from the error status contained in the message, and correct the condition. Then reenter the -RECOVER BSDS command.

System programmer response: The error status portion of this message contains the 2-byte error code (S99ERROR) followed by the 2-byte information code (S99INFO) from the SVC request block. Refer to the appropriate MVS publication for a description of these codes.

DSNJ303I WRITE ERROR ON REPLACEMENT BSDS DSNAME=... ERROR STATUS=...

Explanation: The -RECOVER BSDS command encountered an error while attempting to write to the specified bootstrap data set. Error Status contains the VSAM return and feedback codes. It is a 2-byte field with the first containing the hexadecimal return code and the second containing the hexadecimal feedback code.

System action: Processing of the command is terminated. The system continues in single BSDS mode.

Operator response: Notify the system programmer.

System programmer response: Execute an offline access method services job to delete or rename the replacement BSDS and define a new BSDS with the same name. Reenter the log manager -RECOVER BSDS command to reestablish dual BSDS mode.

DSNJ304I ERROR CLOSING REPLACEMENT BSDS DSNAME=... ERROR STATUS=...

Explanation: The -RECOVER BSDS command encountered an error while attempting to close the specified bootstrap data set. Error Status contains the VSAM return and feedback codes. It is a 2-byte field with the first containing the hexadecimal return code and the second containing the hexadecimal feedback code.

System action: Processing of the command is terminated. The system continues in single BSDS mode.

Operator response: Notify the system programmer.

System programmer response: Execute an offline access method services job to delete or rename the replacement BSDS and define a new BSDS with the same name. Reenter the log manager -RECOVER BSDS command to reestablish dual BSDS mode.

DSNJ305I REPLACEMENT BSDS NOT EMPTY DSNAME=...

Explanation: The -RECOVER BSDS command was issued, but the replacement bootstrap data set was not empty; that is, it contained data.

System action: Processing of the command is terminated. DB2 continues in single BSDS mode.

Operator response: Notify the system programmer.

System programmer response: Execute an offline access method services job to delete or rename the error BSDS and define a new BSDS with the same name. Reenter the log manager -RECOVER BSDS command to reestablish dual BSDS mode.

DSNJ306I DUAL BSDS MODE ALREADY ESTABLISHED

Explanation: The -RECOVER BSDS command was issued, but DB2 was already in dual BSDS mode.

System action: The command is ignored.

DSNJ307I LOG INITIALIZED IN SINGLE BSDS MODE

Explanation: The -RECOVER BSDS command was issued, but the system was initialized in single BSDS mode.

System action: Processing of the command is terminated. The system continues in single BSDS mode.

DSNJ310I (A)SYNCHRONOUS ARCHIVE LOG COMMAND QUIESCE PROCESSING STARTED FOR MAXIMUM OF *xxx* SECONDS

Explanation: An -ARCHIVE LOG command with the MODE(QUIESCE) option has been accepted by the DB2 subsystem. The quiesce processing has commenced.

System action: The DB2 subsystem will attempt to stop all updates to DB2 resources within the time period specified in the message. Users/jobs operating on the DB2 subsystem will be allowed to reach a point of consistency (commit point) before being blocked from further update activity. Jobs and users will be suspended until they are released by DB2 following the initiation of the off-load processing. If the DB2 subsystem can effectively block all users from performing updates before the maximum specified time, then the off-load will be initiated immediately, and normal processing will be resumed.

The message will display either the word SYNCHRONOUS or ASYNCHRONOUS, to inform the user of the type of quiesce processing which has been used.

- If the word SYNCHRONOUS is displayed, then the invoker used the WAIT(YES) option. This means that

quiesce processing will be synchronous to the user; that is, the user can enter additional DB2 commands, but the commands will not be processed by the DB2 command processor until the quiesce processing has ended.

- If the word ASYNCHRONOUS is displayed, then the invoker used the WAIT(NO) option, or did not specify the WAIT option. This means that quiesce processing will be asynchronous to the user; that is, control will be returned to the invoker as soon as the quiesce task has been started. Thus, the DB2 command processor will accept - and process - any new DB2 commands while the quiesce task is running.

This message will be followed by message DSNJ311I or DSNJ317I.

Operator response: No operator response is necessary. However, it can be expected that users and jobs using DB2 resources will be suspended (queued) by DB2 through the duration of the specified time interval, or until the DB2 subsystem can be certain that all update activity has been effectively blocked. At some point, this message will be followed by the DSNJ311I message or DSNJ317I message.

DSNJ311I *csect-name* **ASYNCHRONOUS LOG ARCHIVE (OFFLOAD) TASK INITIATED**

Explanation: A user-initiated -ARCHIVE LOG command has been accepted by the DB2 subsystem. An asynchronous task to archive (off-load) the active log data set has been started.

System action: The current active log data set(s) will be truncated and switched to the next available active log data set(s), and an asynchronous off-load task will be initiated to archive the Active log data set(s).

The term ASYNCHRONOUS is used to indicate that control will be returned to the invoker as soon as the off-load task has been started. Thus, the DB2 command processor will accept — and process — any DB2 commands while the off-load task is running.

This message is followed by message DSNJ312I if the MODE(QUIESCE) option was used with the -ARCHIVE LOG command.

Operator response: The operator should respond per normal operational procedures when the off-load task begins.

DSNJ312I ARCHIVE LOG QUIESCE ENDED. UPDATE ACTIVITY AGAINST DB2 RESOURCES IS NOW RESUMED.

Explanation: An -ARCHIVE LOG command with the MODE(QUIESCE) option was processed by the DB2 subsystem. As part of the MODE(QUIESCE) processing, an attempt was made to stop all new update activity against DB2 resources. This message signals the end of the quiesce processing, and the resumption of normal DB2 activity for all users and jobs which were blocked during the quiesce period.

This message follows message DSNJ311I or DSNJ317I.

System action: The DB2 subsystem has now resumed all normal DB2 activity for all users and jobs that were blocked during the quiesce period.

DSNJ313I PARAMETER SPECIFIED ON THE keyword KEYWORD (value) IS NOT IN THE RANGE OF ALLOWABLE VALUES

Explanation: This message was generated by an error occurring in one of the following commands:

- ARCHIVE LOG command with MODE(QUIESCE)
- SET ARCHIVE
- SET LOG

You specified a value for one of the command options that was outside of the range of allowed values.

System action: Processing for the ARCHIVE LOG, SET ARCHIVE, or SET LOG command is terminated. This message is followed by message DSN9023I.

Operator response: Refer to *DB2 Command Reference* for the correct syntax of the command. Correct the command syntax and reenter the command.

DSNJ314I xxxx KEYWORD REQUIRES yyyy KEYWORD TO ALSO BE SPECIFIED ON THE COMMAND

Explanation: A command was entered which specified the 'xxxx' keyword. However, use of the 'xxxx' keyword requires that the 'yyyy' keyword also be used.

System action: Command processing terminates for the -ARCHIVE LOG or -RECOVER BSDS command. This message is followed by message DSN9023I.

Operator response: Refer to the *DB2 Command Reference* for the correct syntax of the command. Correct the command syntax, and reenter the command.

DSNJ315I -STOP DB2 MODE(FORCE) IN PROGRESS

Explanation: An attempt was made to issue an -ARCHIVE LOG command when a -STOP DB2 MODE(FORCE) command was already in progress.

System action: Command processing terminates for the -ARCHIVE LOG command. The -STOP DB2 MODE(FORCE) processing continues.

DSNJ316I SYSTEM QUIESCE ALREADY IN PROGRESS

Explanation: An attempt was made to issue an -ARCHIVE LOG command with the MODE(QUIESCE) option when a system quiesce was already in progress. The system quiesce could be the result of processing by another -ARCHIVE LOG MODE(QUIESCE) command, or by a -STOP DB2 MODE(QUIESCE) command.

System action: Command processing will terminate. The system quiesce currently in progress will continue.

DSNJ317I ARCHIVE LOG QUIESCE PERIOD EXPIRED. NUMBER OF OUTSTANDING UR'S = xxxx. ARCHIVE LOG PROCESSING WILL BE TERMINATED, AND UPDATE ACTIVITY AGAINST DB2 RESOURCES WILL BE RESUMED. HH:MM:SS

Explanation: An -ARCHIVE LOG MODE(QUIESCE) command was processed by the DB2 subsystem; however, the DB2 subsystem was not able to quiesce all update activity in the user-specified quiesce time interval.

System action: This message is informational only. The DB2 subsystem determined that 'xxxx' units of recovery (URs) did not reach a point of consistency during the quiesce period, and therefore could not be stopped from continuing their associated update processing.

Consequently, the -ARCHIVE LOG processing will be terminated. The current active log data set(s) will not be truncated, and will not be switched to the next available active log data set(s). The log archive (off-load) task will not be created. All jobs and users suspended during the quiesce will be resumed, and normal update activity against DB2 resources will be commenced.

This message will be followed by the DSNJ312I message.

Operator response: Contact the system programmer.

System programmer response: A determination must be made whether the outstanding (nonquiesced) URs represent significant work.

Each user on the system has a unit of recovery if they are modifying DB2 resources. URs are also created by the DB2 subsystem itself for internal processing. Since the purpose of the MODE(QUIESCE) option is to have all URs reach a point of consistency (commit point) before the active log data set is truncated and off-loaded, determine all outstanding nonqueued jobs and users by using -DISPLAY THREAD and the MVS 'D A,L' operator command.

It should be noted that URs may be outstanding due to lock contention between a user or job which holds a resource (and has reached a point of consistency), and a user or job which wants a lock (and therefore cannot reach a point of consistency).

Before resubmitting the -ARCHIVE LOG command with the MODE(QUIESCE) option, either:

- Wait until the threads have been deallocated.
- Wait until the DB2 subsystem is less busy.
- Force the offending thread(s) to terminate.
- Use the TIME option to dynamically override and extend the maximum quiesce time period specified in ZPARMS.
- Or, if having all URs reach a point of consistency in the active log is no longer critical, issue the -ARCHIVE LOG command without the MODE(QUIESCE) option.

Important: If the decision is made to use the -ARCHIVE LOG command without the MODE(QUIESCE) option, the active log data set(s) will be truncated without regard to quiescing activity on the DB2 subsystem. If the resulting archive log data set is used for recovery, it is possible that some URs may be found to be In-flight, In-Abort, In-Commit, or In-doubt during DB2 initialization.

If expiration of the quiesce period before all URs reach a consistent point are a chronic problem, an adjustment to ZPARMs may have to be made. The quiesce period is controlled by the 'Maximum Quiesce Period' ZPARM. Refer to the discussion about the Archive Log Parameters (DSNTIPA) panel in Part 2 of *DB2 Installation Guide*.

DSNJ318I ARCHIVE LOG COMMAND ALREADY IN PROGRESS

Explanation: An attempt was made to issue an ARCHIVE LOG command when an ARCHIVE LOG command was already in progress. In data sharing, the ARCHIVE LOG command currently in progress could be initiated by another member.

Operator response: If the conflicting command satisfies your archival request, do not reissue the command. Otherwise, wait for the conflicting command to complete and then reissue the -ARCHIVE LOG command.

DSNJ319I *csect name* **CURRENT ACTIVE LOG DATA SET IS THE LAST AVAILABLE ACTIVE LOG DATA SET. ARCHIVE LOG PROCESSING WILL BE TERMINATED.**

Explanation: The -ARCHIVE LOG command was rejected because the current active log is the last available active log data set. To process the command when these conditions exist would cause the DB2 subsystem to exhaust its available active log resources

and immediately halt processing.

System action: Processing for the command is terminated. If the situation is not corrected, the DB2 subsystem will issue the DSNJ110E message (if it has not already done so) as the available active log data space reaches critically low levels. Ultimately, message DSNJ111E will be issued when the available active log data space is exhausted, and the DB2 online processing will halt until active log space is made available.

User response: Notify the operator or system programmer.

Operator response: To clear this condition, steps must be taken to complete other waiting off-load tasks. Once another active log is made available (re-usable) by completing the off-load process for it, the command processing for the current active log can proceed.

Execute an MVS *Display System Requests* command (D R,L for example) to determine the outstanding requests related to the log off-load process. Take the necessary action to satisfy any requests, and permit off-load to continue. If off-load fails to complete normally, or cannot be initiated, notify the systems programmer.

System programmer response: Either correct the problem that is causing the off-load process to fail, or add more active log data sets. Note that the latter action requires DB2 to be down and the change log inventory (DSNJU003) utility to be run.

Possible causes for the shortage of active log data space are:

- Excessive logging. For example, too many users requesting logs at the same time, or too many utilities specifying LOG (YES).
- Delayed or slow off-loading. For example, failure to mount archive volumes, incorrect replies to off-load messages, or slow device speeds.
- Excessive use of the -ARCHIVE LOG command. Each invocation of the command causes the DB2 subsystem to switch to a new active log data set. Excessive use could consume the available active log data space if the resulting off-loads were not processed in a timely manner.
- Off-loads(s) failed (see Operator Response).
- Insufficient active log space (see Operator Response).

DSNJ320I *csect-name* **UNABLE TO PROCESS LOG TRUNCATION REQUEST DUE TO SUBSYSTEM COMPONENT** *action* **FAILURE**

Explanation: While processing an -ARCHIVE LOG command, an internal request was made of the log buffer output routine to force-write the log buffers and to truncate and switch the active log to the next available active log data set(s). The request failed during the processing of the 'action' specified in the message.

System action: Processing for the command is terminated.

User response: Notify the system programmer.

Operator response: Notify the system programmer.

System programmer response: This error is an internal error detected by the DB2 subsystem. The error may be caused by an unrelated ABEND in the Log Buffer Writer component (DSNJWxxx), by a -STOP DB2 MODE(FORCE) command, or by abnormal subsystem termination. Refer to messages and ABENDS which precede this message.

DSNJ321I UNABLE TO CONTINUE ARCHIVE LOG QUIESCE DUE TO INTERNAL SUBSYSTEM ERROR. ARCHIVE LOG PROCESSING WILL BE TERMINATED, AND UPDATE ACTIVITY AGAINST DB2 RESOURCES WILL BE RESUMED.

Explanation: An -ARCHIVE LOG command with the MODE(QUIESCE) option was processed by the DB2 subsystem. As part of the MODE(QUIESCE) processing, an attempt was made to stop all new update activity against DB2 resources. During the internal subsystem interaction between the log manager and the recovery manager, an internal interface error was detected by the recovery manager.

System action: The -ARCHIVE LOG MODE(QUIESCE) processing is terminated. This message will be followed by message DSNJ312I after all users and jobs quiesced by the MODE(QUIESCE) processing are resumed.

User response: Notify the system programmer.

Operator response: Notify the system programmer.

System programmer response: This error is an internal error detected by the DB2 subsystem. Retry the -ARCHIVE LOG MODE(QUIESCE) command. If the error persists, the active log data sets may be switched using the -ARCHIVE LOG command without the MODE(QUIESCE) option.

DSNJ322I DISPLAY ARCHIVE REPORT FOLLOWS—

Explanation: This message is always generated as the title for the -DISPLAY ARCHIVE command report output. It precedes the other messages generated by this command.

The series of messages DSNJ323I through DSNJ329I provide the formatted report for -DISPLAY ARCHIVE.

DSNJ323I COUNT TIME

Explanation: The column headings for the list of parameter specifications generated by the -DISPLAY ARCHIVE command are displayed. The message

number is not included in this message.

DSNJ324I (TAPE UNITS) (MIN,SEC)

Explanation: The format of the COUNT and TIME parameter specifications displayed by the -DISPLAY ARCHIVE command are shown. The message number is not included in this message.

DSNJ325I name count time

Explanation: This message, in combination with messages DSNJ323I and DSNJ324I, provides the parameter information that is a part of the -DISPLAY ARCHIVE report.

name Identifies the count and time values in the message line.

DSNZPARM
Values specified during DB2 installation.

CURRENT
Current values in the DB2 system.

count The value for COUNT as specified in name.

time The value for TIME as specified in name.

The message number is not included in this message.

DSNJ326I

Explanation: This message separates messages DSNJ325I and DSNJ327I that are generated by the -DISPLAY ARCHIVE command. The message number is not included in this message.

DSNJ327I ADDR STATUS CORR-ID VOLSER DATASET_NAME

Explanation: This message displays the column headings for the DISPLAY ARCHIVE report.

The message number is not included in this message.

DSNJ328I tapeaddr status corr-id vname dsname

Explanation: This message, in combination with message DSNJ327I, provides the tape unit information that is part of the -DISPLAY ARCHIVE report.

tapeaddr
The physical tape address of the tape unit allocated to read archive log.

status Status of tape address:

BUSY Tape address is actively processing an archive log data set.

PREM Tape address is active and allocated for premount.

AVAIL Tape unit is inactive and waiting for work.

corr-id The correlation id associated with the user of the tape being processed. If the status is PREM or AVAIL, this column might contain '*****', indicating no current user is associated with this tape.

vname The name of the tape volume that is mounted on the tape address.

dsname
The data set on the tape volume that is being processed (BUSY and PREM) or was last processed (AVAIL).

The message number is not included in this message.

DSNJ329I END OF DISPLAY ARCHIVE REPORT.

Explanation: This message follows messages DSNJ322I through DSNJ328I and indicates that the -DISPLAY ARCHIVE report was generated successfully. The message number is not included in this message.

DSNJ330I ARCHIVE LOG VOLUMES REQUIRED FOR CONNECTION ID conn-id CORRELATION ID corr-id

Explanation: This message is the title for the ARCHIVE LOG VOLUMES report output. The ARCHIVE LOG VOLUMES report lists the archive log volumes needed by the correlation ID for that connection ID. This message is generated automatically by the archive read process at the first archive log tape mount for corr-id. This message precedes the other messages generated by this action.

This message, in combination with messages DSNJ331I and DSNJ332I, provides the formatted report for the ARCHIVE LOG VOLUMES requested.

conn-id
An identifier representing the 'connection name' used to establish the thread.

corr-id An identifier associated with a specified thread. In TSO, it is either the authorization ID or the job name.

DSNJ331I avolume, avolume, avolume, avolume ...

Explanation: This message provides the tape volume information for the Archive Log Volumes report output.

avolume
The name of the archive log volume required for this correlation ID. An 'avolume' prefixed with an '*' signifies that the data on the archive log volume is also mapped by an active log data set. As such, the 'avolume'

might not be required for the read process, since the data is read from the active log, if possible.

DSNJ332I END OF ARCHIVE LOG VOLUMES REPORT

Explanation: This message follows messages DSNJ330I and DSNJ331I and indicates that the ARCHIVE LOG VOLUMES report was generated successfully. The message number is not included in this message.

DSNJ333I *csect-name* **SYSTEM CHECKPOINT INITIATED**

Explanation: This message indicates that the SET LOG command processed successfully and that a system checkpoint has been initiated.

DSNJ334I PARAMETER SPECIFIED ON THE keyword KEYWORD (VALUE) CONTAINS INVALID CHARACTERS

Explanation: A -SET ARCHIVE command was processed by the DB2 subsystem. One of the command keywords specified a value that contains an invalid character.

System action: Processing for the -SET ARCHIVE command is terminated. This message is followed by the DSN9023I message.

Operator response: Refer to *DB2 Command Reference* for the syntax of the command. Correct the syntax and reenter the command.

DSNJ335I INVALID KEYWORDS COMBINATION IN SET ARCHIVE COMMAND

Explanation: This command requires that keyword COUNT or keyword TIME be specified. If both are omitted, the keyword DEFAULT must be specified. It is invalid to specify keyword DEFAULT when either the COUNT keyword or TIME keyword is specified.

System action: Processing for the -SET ARCHIVE command is terminated. This message is followed by the DSN9023I message.

Operator response: Refer to *DB2 Command Reference* for the syntax of the command. Correct the syntax and reenter the command.

DSNJ336I UNABLE TO ALLOCATE TAPE UNIT FOR conn-id corr-id, x ALLOCATED, y ALLOWED.

Explanation: An attempt to allocate a tape unit for connection ID 'conn-id' failed. The current maximum tape unit specified is 'y', but only 'x' are physically available.

System action: The 'conn-id corr-id' process is queued until either an allocated tape unit becomes free or more tape units are varied online and made available to the DB2 archive read task.

Operator response: This situation rectifies itself over time as currently allocated tape units become available. To improve throughput, vary additional tape units online and make them available to DB2. Note that a queue rescan is not attempted until the -SET ARC READ COUNT command is issued or when an allocated tape dismounts.

DSNJ337I SET ARCHIVE COMMAND COMPLETED COUNT(count) TIME(minute,second)

Explanation: This message indicates that the -SET ARCHIVE command processed successfully and displays the current values for COUNT and TIME.

count The value for the COUNT parameter

minute The 'minute' value for the TIME parameter

second The 'second' value for the TIME parameter

DSNJ338I NO ARCHIVE READ ACTIVITY

Explanation: This message indicates that there is no
current archive read activity to report.

The message number is not included in this message.

DSNJ339I *csect-name* **SET LOG COMMAND COMPLETED,** *type***(***quantity***)**

Explanation: This message indicates that the SET LOG command processed successfully, and displays the current checkpoint frequency. The checkpoint frequency remains in effect until DB2 is restarted or until the value is modified by a SET LOG or SET SYSPARM command.

type Along with *quantity*, determines the current system checkpoint frequency. It indicates MINUTES for a time-based checkpoint frequency, or LOGRECORDS if the checkpoint frequency is determined by the number of log records that are written.

quantity
The number of minutes, or the number of log records that determine the current checkpoint frequency.

DSNJ340I *csect-name* **COMMAND KEYWORD** *xxx* **IS ONLY VALID FOR DATA SHARING**

Explanation: An attempt was made to issue a DB2 command with KEYWORD *xxx* on a DB2 that was not initialized to participate in data sharing.

System action: The command request is rejected.

DSNJ341I *csect-name* **-ARCHIVE LOG COMMAND NOT ALLOWED, LOGGING IS SUSPENDED**

Explanation: An attempt to issue an -ARCHIVE LOG command failed because a -SET LOG SUSPEND command has suspended logging.

System action: Command processing terminates for the -ARCHIVE LOG command. You may reissue the command when logging has been resumed by the -SET LOG RESUME command.

DSNJ350I *csect-name* **(A)SYNCHRONOUS ARCHIVE LOG COMMAND QUIESCE PROCESSING STARTED IN DATA SHARING**

Explanation: The DB2 subsystem of a data sharing group accepted an ARCHIVE LOG command with the MODE(QUIESCE) option.

Each active DB2 member in the data sharing group performs the quiesce operation. The member originating the quiesce issues message DSNJ350I; all other members issue DSNJ310I as they start quiescing.

DSNJ351I *csect-name* **MEMBER** *member name* **STATUS=***member status*

Explanation: This message is one of a list of elements
that display archive related status of the indicated
member of the data sharing group. These member
status messages are preceded by the message DSNJ359I
and are followed by DSNJ361I.

member name identifies the member of the data sharing group.

member status indicates archive related status of the named member. It appears as one of the following:.

ABNORMALLY TERMINATING
The named member was undergoing abnormal subsystem termination at the time ARCHIVE LOG processing was active for the data sharing group. ARCHIVE LOG command processing for the remainder of the data sharing group terminates unsuccessfully.

ARCHIVE ALREADY ACTIVE
ARCHIVE LOG command processing was running at the named member at the time a succeeding ARCHIVE LOG command was issued. The succeeding ARCHIVE LOG command terminates unsuccessfully for the remaining members of the data sharing group.

CANCELED
ARCHIVE LOG command processing for the named member was canceled due to a problem encountered elsewhere in the data sharing group. The quiesce operation is interrupted and terminated and update activity against DB2 resources is resumed before command processing is terminated for the member.

COMMUNICATIONS FAILURE
A communications failure occurred between the named member and one or more other members in the data sharing group. The quiesce operation is interrupted and update activity against DB2 resources is resumed before command processing is terminated for the member. ARCHIVE LOG command processing for the remainder of the data sharing group terminates unsuccessfully.

FAILED
The named member was inactive at the time the ARCHIVE LOG command was issued. Termination of the DB2 subsystem for this member was abnormal. This status, if displayed for any member, will cause ARCHIVE LOG command processing for the entire data sharing group to terminate unsuccessfully.

NOT AVAILABLE
Archive command status for the named member could not be determined. This condition typically occurs when the named member fails to respond during ARCHIVE LOG command processing and is usually accompanied by message DSNJ353I and/or message DSNJ355I.

QUIESCED
The named member was active at the time the ARCHIVE LOG command was issued. Update activity against DB2 resources had successfully quiesced for that member (no active URs) in response to specifying MODE(QUIESCE).

QUIESCED/OFFLOADING
The named member was active at the time the ARCHIVE LOG command was issued. Update activity against DB2 resources had successfully quiesced for that member (no active URs) in response to specifying MODE(QUIESCE). The active log at the time was truncated and an offload process was initiated after all members in the data sharing group had reached a quiesce point.

STOPPED
The named member was inactive at the time the ARCHIVE LOG command was issued. Termination of the DB2 subsystem for this member was normal with no incomplete URs outstanding. Log truncation and offload processing are scheduled to be performed automatically the next time the named member is started. When the member is started, it is important to ensure that archive processing occurs automatically. Certain error conditions might require entering an

ARCHIVE LOG command via the operator console to cause log truncation and offload processing to be performed. This member status is accompanied by warning message DSNJ362A.

TIME EXCEEDED
The named member could not quiesce its update activity within the allotted period of time. The incomplete quiesce operation is interrupted and update activity is resumed before command processing is terminated for the member. ARCHIVE LOG command processing for the remainder of the data sharing group is unsuccessfully terminated.

DSNJ352W *csect-name* **MEMBERS WERE NOT OFFLOADED**

Explanation: DB2 issues this message during processing of an ARCHIVE LOG command with the MODE(QUIESCE) option. One or more quiesced members were not available to receive the archive request. DSNJ351I messages follow to display the quiesced members.

At the next start up of the quiesced member, DB2 switches (truncates) the current active log and initiates the archive task.

DSNJ353I *csect-name* **DSNJ353I MEMBER** *member name* **NOT RESPONDING.**

Explanation: This message is displayed when a response is needed but not received within an expected period of time during ARCHIVE LOG MODE(QUIESCE) command processing. This can occur as a result of abnormal subsystem termination under the named member or, possibly from communications problems among members of the data sharing group.

Operator response: Examine the console log of the named member for operator information messages that might relate to the problem.

DSNJ354I *csect-name* **DSNJ354I ARCHIVE LOG: ALL ACTIVE MEMBER(S) STARTED OFFLOAD TASKS**

Explanation: This message is displayed when MODE(QUIESCE) is specified with the ARCHIVE LOG command, after all active members in the data sharing group have successfully quiesced their update activity within the allotted time period. For each active member in the data sharing group, the active log is truncated and switched, and an offload task is initiated to begin the offload process. Log truncation, switch and offload processing for stopped (and quiesced) members is performed the next time they are started.

DSNJ355I *csect-name* **ARCHIVE LOG: ONE OR MORE ACTIVE MEMBERS FAILED TO RESPOND OR START OFFLOAD TASKS**

Explanation: This message is displayed when MODE(QUIESCE) is specified with the ARCHIVE LOG command, and one or more active members could not successfully quiesce their update activity in the time allotted. Subsystem activity already quiesced is resumed, and ARCHIVE LOG command processing terminates unsuccessfully for all active members in the data sharing group. Message DSNJ353I might also appear with this message.

Operator response: See the list of DSNJ351I member status messages following message DSNJ359I to determine the reason for the archive command failure.

DSNJ356I *csect-name* **COMMUNICATIONS FAILURE TERMINATED COMMAND**

Explanation: This message appears as a result of the failure of a communications request by one or more members of the data sharing group. This message is likely to be accompanied by DSN7xxx or other error messages and/or an SVC dump.

System action: ARCHIVE LOG command processing terminates unsuccessfully for all members in the group.

DSNJ357E *csect-name* **SYSTEM QUIESCE FAILED**

Explanation: This message indicates the unsuccessful outcome of an ARCHIVE LOG MODE(QUIESCE) command. This message indicates that one or more members were unable to quiesce their update activity in the time allowed. It is usually accompanied by one or more related messages (e.g., DSNJ355I), and a list of member status messages indicating the reason for the failure.

Operator response: Refer to messages DSNJ359I, DSNJ351I and DSNJ361I to determine archive related status of members in the data sharing group.

DSNJ359I **MEMBER RESPONSES**

Explanation: This message marks the beginning of a list of one or more DSNJ351I member status messages and precedes the first status message in the list.

DSNJ361I *csect-name* **END OF MEMBER RESPONSES LIST**

Explanation: This message marks the end of a list of one or more DSNJ351I member status messages and follows the last status message in the list.

DSNJ362E *csect-name* **WARNING: LOGS OF** *nn* **STOPPED OR FAILED MEMBERS NOT OFFLOADED.**

Explanation: This message is displayed when either MODE(QUIESCE) or SCOPE(GROUP) is specified with the ARCHIVE LOG command, and one or more members of the data sharing group either stopped or failed and were not available to receive the archive request. Active log truncation, log switch and offload processing for these inactive members is scheduled to be performed the next time those members are started.

Operator response: Verify that log switch and offload processing does automatically occur when the inactive members are started. Certain error conditions might require the ARCHIVE LOG command to be individually issued for a member if ARCHIVE LOG processing does not occur automatically. Refer to messages DSNJ359I, DSNJ351I, and DSNJ361I to identify the inactive members.

DSNJ370I *csect-name* **LOG DISPLAY CURRENT COPY1 LOG =** *dsname1* **IS** *pct*% **FULL CURRENT COPY2 LOG =** *dsname2* **IS** *pct*% **FULL H/W RBA =** *hw-rba*, **H/O RBA =** *ho-rba* **FULL LOGS TO OFFLOAD =** *nn* **OF** *mm*, **OFFLOAD TASK IS** *status*

Explanation: This message, in combination with message DSNJ371I, provides the output from the DISPLAY LOG command.

dsname1
The data set name of the current active log copy1 data set. If this copy of the log is not currently active, dsname1 will display INACTIVE.

dsname2
The data set name of the current active log copy2 data set. If this copy of the log is not currently active, dsname2 will display INACTIVE.

pct The percentage of log space within the active log data set that has been written.

hw-rba The RBA of the most recently written log record.

ho-rba The highest log RBA that has been offloaded to an archive log data set.

nn The number of full active log data sets that have not been archived.

mm The total number of active log data sets.

status Status of the offload task:

BUSY,ALLC
Task is busy, allocating an archive log data set. This could indicate a tape mount request is outstanding.

BUSY,BSDS
Task is busy, copying the BSDS data set.

BUSY,ACTV
Task is busy, copying an active log data set.

BUSY Task is busy, other processing.

AVAILABLE
Task is available for work.

DSNJ371I **DB2 RESTARTED** *time-date-1* **RESTART RBA** *rstrt-rba* **CHECKPOINT FREQUENCY** *quantity type* **LAST SYSTEM CHECKPOINT TAKEN** *time-date-2*

Explanation: This message, in combination with message DSNJ370I, provides the output from the DISPLAY LOG command.

time-date-1
The time and date when DB2 started.

rstrt-rba
The RBA that DB2 began logging when started.

quantity
The number of minutes, or the number of log records that determine the current checkpoint frequency.

type Along with quantity, determines the current system checkpoint frequency. It indicates MINUTES for a time-based checkpoint frequency, or LOGRECORDS if the checkpoint frequency is determined by the number of log records that are written.

time-date-2
The time and date of the last system checkpoint.

DSNJ372I *csect-name* **UPDATE ACTIVITY HAS BEEN SUSPENDED FOR** *aaaa* **AT RBA** *xxxxxxxxxxxx*, **LRSN** *yyyyyyyyyyyy*, **PRIOR CHECKPOINT RBA** *zzzzzzzzzzzz*

Explanation: This message indicates that the -SET LOG command with the SUSPEND option has completed successfully.

System action: All log and database update activity has been suspended for DB2 subsystem *aaaa*. *xxxxxxxxxxxx* contains the RBA of the last record written to the log. In a data sharing environment, *yyyyyyyyyyyy* contains the LRSN of the last record written to the log. Otherwise, *yyyyyyyyyyyy* contains the high written RBA. *zzzzzzzzzzzz* contains the RBA of the last system checkpoint.

Operator response: Issue the -SET LOG command

with the RESUME keyword when ready to resume update activity.

DSNJ373I *csect-name* **UPDATE ACTIVITY HAS BEEN RESUMED FOR** *xxxx*

Explanation: This message indicates that the -SET LOG command with the RESUME option has completed successfully.

System action: All log and database update activity has been resumed for DB2 subsystem *xxxx*.

DSNJ374I *csect-name* **SET LOG** *options* **COMMAND FAILED,** *reason*

Explanation: A -SET LOG command has failed. *options* were specified on the command. *reason* describes the cause of the failure:

ALREADY SUSPENDED
A -SET LOG SUSPEND command was issued while update activity was suspended by a previous command.

DB2 STOPPING
A -SET LOG SUSPEND command was issued, but DB2 is stopping.

QUIESCE ACTIVE
A -SET LOG SUSPEND command was issued, but a system quiesce that was initiated by an -ARCHIVE LOG command is in progress.

NOT SUSPENDED
A -SET LOG RESUME command was issued, but update activity is not suspended.

COMMAND ABEND
A -SET LOG command abnormally terminated. Update activity is not suspended.

LOCK TIMEOUT
A lock timeout has occurred. Update activity is not suspended.

System action: Command processing terminates.

DSNJ401I *csect-name rrrr* **RECORD NOT FOUND**

Explanation: An attempt was made to read the 'rrrr' record from the BSDS. In so doing, the read routine (DSNJU01B) could not find the record.

System action: Utility processing continues. This message is written to the SYSPRINT data set, not to a console.

DSNJ402I UNABLE TO LOCATE aaaa TBLSPACE NAME IN SYSTEM CATALOG OR DIRECTORY DATABASES

Explanation: The table space name specified on the TBLSPACE parameter of the change log inventory (DSNJU003) utility SYSTEMDB statement could not be found in any DB2 subsystem catalog or directory database. The table space name in error is named in the 'aaaa' message keyword.

System action: All subsequent DSNJU003 utility statements are bypassed, and DSNJU003 processing is terminated with a return code of 4. The DB2 SYSTEMDB record is not updated.

This message is written to the SYSPRINT data set, not to a console.

User response: Correct the table space name specified on the TBLSPACE parameter, and resubmit the change log inventory (DSNJU003) utility job.

DSNJ403I CAPACITY OF BSDS SYSTEMDB RECORD EXCEEDED

Explanation: The BSDS SYSTEMDB record contains VSAM passwords for data sets in the DB2 catalog and directory, and has a maximum capacity of 168 entries. This message is issued when the number of eligible DB2 catalog and directory data sets exceeds the maximum capacity of the BSDS SYSTEMDB record. This is an internal error.

System action: All subsequent change log inventory (DSNJU003) utility statements are bypassed, and DSNJU003 processing is terminated with a return code of 4. The DB2 SYSTEMDB record is not updated.

This message is written to the SYSPRINT data set, not to a console.

User response: Notify the systems programmer.

DSNJ404E keyword NOT ALLOWED FOR operation OPERATION

Explanation: An invalid keyword was used during the operation named 'operation'.

System action: The current utility processing is terminated. This message is written to the SYSPRINT data set, not to a console.

User response: Correct the control statement and rerun the utility.

DSNJ405E keyword1 KEYWORD MUTUALLY EXCLUSIVE WITH keyword2 KEYWORD

Explanation: 'keyword1' and 'keyword2' are mutually exclusive and cannot appear on the same control statement.

System action: The current utility processing is terminated. This message is written to the SYSPRINT data set, not to a console.

User response: Determine the extraneous keyword and remove it from the control statement. Rerun the utility.

DSNJ406E EITHER keyword1 KEYWORD OR keyword2 KEYWORD MUST BE SPECIFIED

Explanation: A required keyword was not used on the control statement. Use either 'keyword1' or 'keyword2' with the attempted control statement type.

System action: The current utility processing is terminated. This message is written to the SYSPRINT data set, not to a console.

User response: Correct the control statement and rerun the utility.

DSNJ407I *csect-name* **NO VALID CHECKPOINT RBA FOUND. LOG WILL BE SCANNED AT RESTART**

Explanation: After completing its search through the resource manager status table and the checkpoint queue, no valid checkpoint RBA was found within the specified range.

System action: Utility processing continues, and will set a return code of 4. This message is written to the SYSPRINT data set, not to a console.

User response: The last 100 checkpoints are recorded
in the BSDS, including the log STARTRBA and log
ENDRBA of the checkpoint range. When either
STARTRBA or ENDRBA/ENDLRSN are specified, the
change log inventory (DSNJU003) utility attempts to
locate a valid checkpoint in the range. In this case,
DSNJU003 was unsuccessful in finding a valid
checkpoint. When DB2 is restarted, the log will be
scanned to locate the checkpoint prior to the log
truncation point.

DSNJ408I *csect-name* **CHECKPOINT RBA FOUND, RBA =** *rba*, **TIME =** *time*

Explanation: After completing its search through the resource manager status table and the checkpoint queue, 'rba' was the most recent checkpoint RBA in the specified range and 'time' was the time of the checkpoint.

System action: Utility processing continues.

DSNJ409I UNABLE TO WRITE ACTIVE LOG RECORD TO THE BSDS DATASET. DDNAME=xxxxxx

Explanation: A program encountered an I/O error when attempting to write a BSDS active log record to the BSDS data set. 'xxxxxx' is the DDNAME of the BSDS data set being accessed.

System action: The program will terminate.

Operator response: Notify the system programmer.

System programmer response: One or more error messages describing the specific error preceded this message. This error is considered unrecoverable and will leave the BSDS data set in an uncertain state. It will be necessary to either: recreate the data set using access method service routines and then use change log inventory (DSNJU003) to make the active and archive log data set(s) known to the BSDS, or recover the BSDS using a backup copy.

DSNJ410E *csect-name* **I/O ERROR DURING** *nnn* **PROCESSING OF** *yyy* **RECORD**

Explanation: An I/O error occurred during a READ or WRITE of a record. 'nnn' is either READ or WRITE, depending on the operation. 'yyy' specifies the Conditional Restart control record, checkpoint queue record, archive log command history record, or resource manager status table.

System action: The current utility processing is terminated. This message is written to the SYSPRINT data set, not to a console, and will be accompanied by message DSNJ212E or DSNJ213E.

User response: Determine the cause of the I/O error based upon the error status information provided in messages DSNJ212E or DSNJ213E.

DSNJ411I *csect-name* **CRESTART** *nnnn* **FOR CRCRID =** *yyyy*, **DDNAME =** *ddname*

Explanation: A CRESTART request has just completed. 'nnnn' is either CREATE or CANCEL. 'yyyy' is the restart control record hexadecimal identifier and 'ddname' is the BSDS data set (SYSUT1 or SYSUT2) associated with the request.

System action: Current utility processing continues. This message is written to the SYSPRINT data set, not to a console.

User response: Note the record identifier for future reference.

DSNJ412E *csect-name* **RESTART CONTROL RECORD NOT FOUND IN BSDS**

Explanation: The CRESTART CANCEL keyword was specified but the conditional restart control record does not exist in the BSDS data set.

System action: Current utility processing is terminated. This message is written to the SYSPRINT data set, not to a console.

User response: CREATE the conditional restart control record prior to attempting the CANCEL function.

DSNJ413E *csect-name* **INVALID LOG RANGE SCOPE OR CHECKPOINT SPECIFIED**

Explanation: The values specified thru the

STARTRBA, ENDRBA, and CHKPTRBA keywords are invalid.

System action: Current utility processing is terminated. This message is written to the SYSPRINT data set, not to a console.

User response: Ensure that the log range values are correct and correspond to the other log range values either specified or defaulted. A valid log range is where STARTRBA is less than or equal to CHKPTRBA, and CHKPTRBA is less than ENDRBA. The STARTRBA most be less than or equal to the ENDRBA.

DSNJ414I *csect-name* **COLD START WILL RESULT FROM THIS RESTART CONTROL RECORD. FORWARD AND BACKOUT SET TO NO.**

Explanation: STARTRBA and ENDRBA are equal. A cold start will result if this restart control record is used during DB2 restart. No forward or backout processing will be performed.

System action: CRESTART processing continues.

User response: No additional actions are required if a cold start of the system is desired. If a cold start is not desired, reissue the CRESTART and either CANCEL the current restart control record, or CREATE a new restart control record.

DSNJ415E *csect-name* **ENDRBA =** *rba* **IS INVALID, MUST BE A MULTIPLE OF 4K**

Explanation: The specified ENDRBA at 'rba' is not a multiple of 4KB.

System action: CRESTART processing is terminated.

User response: Correct the ENDRBA value on the CRESTART statement and rerun the utility.

DSNJ416I **WARNING - BSDS UTILITY TIMESTAMP MISMATCH DETECTED. PROCESSING CONTINUES.**

Explanation: As a result of a change log inventory (DSNJU003) update, it was discovered that the SYSUT1 BSDS and SYSUT2 BSDS timestamps are unequal. Being unequal indicates the possibility of a BSDS mismatch.

System action: Current utility processing continues.

User response: Run the print log map (DSNJU004) utility against the SYSUT1 BSDS and SYSUT2 BSDS. Determine if each BSDS is current. If each BSDS is current, this warning can be ignored. If either BSDS is not current, delete the obsolete data set and define a replacement data set, then copy the current BSDS into the replacement data set.

DSNJ417I **REQUIRED** *xxxxxxxx* **PARAMETER FOR** *yyyyyyyy* **OPERATION IS MISSING**

Explanation: A required parameter 'xxxxxxxx' for a Log utility operation was missing from the Log utility control statement. The attempted operation is 'yyyyyyyy'.

System action: The Log utility 'yyyyyyyy' operation does not perform its function. All subsequent Log utility control statements are processed. A nonzero return code is issued by the utility.

DB2 writes this message to the SYSPRINT data set, not to a console.

User response: Add the missing parameter to the control statements associated with the specified operation and rerun the utility.

DSNJ418I **WARNING - A NOTREUSEABLE ACTIVE LOG DATA SET WAS DELETED FROM BSDS LOG INVENTORY, STARTRBA=...,ENDRBA=...**

Explanation: The data set name specified on the DSNAME parameter of the change log inventory (DSNJU003) utility DELETE statement was a NOTREUSABLE active log.

System action: DSNJU003 processing continues, it will terminate with a return code of 4. This message is written to the SYSPRINT data set, not to a console.

User response: No additional actions are required if a deletion of a NOTREUSABLE active log was desired. If this was not desired, recreate the deleted log by using the NEWLOG statement with the RBA values specified in the warning message.

DSNJ419I **UNABLE TO WRITE ARCHIVE LOG RECORD TO THE BSDS DATASET. DDNAME=***xxxxxx*

Explanation: A program encountered an I/O error when attempting to write a BSDS archive record to the BSDS data set. 'xxxxxx' is the DDNAME of the BSDS data set being accessed.

System action: The program will terminate.

Operator response: Notify the system programmer.

System programmer response: One or more error messages describing the specific error preceded this message. This error is considered unrecoverable and will leave the BSDS data set in an uncertain state. It will be necessary to either: recreate the data set using access method service routines and then use change log inventory (DSNJU003) utility to make the active and archive log data set(s) known to the BSDS, or recover the BSDS using a backup copy.

DSNJ420I UNABLE TO WRITE BCR1 RECORD TO BSDS DATASET. DDNAME=*xxxxxx*

Explanation: A program encountered an I/O error when attempting to write the BSDS control record to the BSDS data set. 'xxxxxx' is the DDNAME of the BSDS data set being accessed.

Operator response: Notify the system programmer.

System programmer response: One or more error messages describing the specific error preceded this message. Refer to the specific message(s) for error analysis and the appropriate action to take.

DSNJ421I THE BSDS IS NOT AT THE PROPER RELEASE LEVEL. DDNAME=*xxxxx*

Explanation: This message is issued when a program attempting to access The BSDS data set is not at a compatible release level. This can be one of the following:

- The program accessing the BSDS data set is at a NEW release level and the BSDS is at the old level.
- The program accessing the BSDS data set is at an OLD release level and the BSDS is at the NEW level.
- The program encountered an UNKNOWN release level on the BSDS data set.

System action: The program will terminate with no action taken.

System programmer response: If the program that encountered the error was one of the BSDS conversion programs (DSNJMIG0 or DSNJFAL0) an unrecoverable error was found on the BSDS data set and it must be recreated. If the error was encountered running some other utility program (i.e., change log inventory or print log map), one of the following actions may be taken:

- Run a version of the program that is compatible with the release level of the BSDS.
- Run the appropriate formatting program (DSNJMIG0 or DSNJFAL0) to convert the BSDS data set to a release level compatible with the running program. If this was successful, rerun the original program.

DSNJ422I LOG RECORDS HAVE ALREADY BEEN FORMATTED - THIS UTILITY WILL TERMINATE

Explanation: This message will occur when one of the BSDS formatting programs detects the BSDS had already been re-formatted or was already at the desired format level.

System action: The program will terminate without any action being taken.

System programmer response: If the appropriate BSDS formatting program was being run, no further action is necessary.

DSNJ423I ACTIVE LOG MAY NOT HAVE BEEN ARCHIVED. DSN=*xxxxxx*

Explanation: The program was attempting to convert an old BSDS data set with more than 31 active log data sets to the new format. The program will drop the oldest active log data set(s) to get to the new maximum (31) limit. When dropping one of the old active log data sets from the BSDS, it was found to have been flagged as NOT REUSABLE (i.e., this data set MAY NOT have been archived). 'xxxxxx' will contain the data set name of the active log in question.

System action: The program will remove this active log data set from the BSDS and continue.

System programmer response: Check to see if this data set had been archived. The active log data set was not deleted on disk but removed only from the BSDS data set, so it may still be archived, if needed. Use access method service routines to copy the VSAM active log data set to a sequential archive log data set.

DSNJ424I *csect-name* INVALID CONTROL STATEMENT. THE SECOND WORD MUST BE 'DDNAME' OR '*' OR MUST START WITH A PARENTHESIS.

Explanation: The control statement is invalid.

System action: The control statement is ignored. DB2 uses the SYSUT1 DD statement to locate the BSDS to be mapped. If the SYSUT1 DD statement is not available, the utility terminates.

User response: Refer to *DB2 Utility Guide and Reference* for information on creating a print log map (DSNJU004) utility control statement. Correct or remove the control statement and rerun the job.

Problem determination: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 32.

DSNJ425I INVALID *xxxxxx* VALUE/FORMAT FOR *yyyyyyyy* PARAMETER (YYYYDDDHHMMSST)

Explanation: 'xxxxxx' is the field in the 'yyyyyyyy' parameter that contains an incorrect value or incorrect format.

System action: The current utility is terminated. This message is written to the SYSPRINT data set, not to a console.

User response: Correct the control statement and rerun the utility.

DSNJ426I ENDTIME VALUE CANNOT BE LESS THAN STARTIME VALUE

Explanation: The 'STARTIME' and 'ENDTIME' parameters specify a time range. Hence, the ENDTIME value must be equal to or greater than STARTIME value.

System action: The current utility is terminated. This message is written to the SYSPRINT data set, not to a console.

User response: Correct the control statement and rerun the utility.

DSNJ427I CHECKPOINT RECORD ADDED TO QUEUE

Explanation: The checkpoint record specified has been added to the checkpoint queue.

System action: The BSDS checkpoint queue is modified as stated. This message is written to the SYSPRINT data set, not to a console.

DSNJ428I CHECKPOINT RECORD DELETED FROM QUEUE, STARTRBA=*startrba*, ENDRBA=*endrba*

Explanation: The checkpoint record specified has been deleted from the Checkpoint Queue. 'startrba' and 'endrba' was the RBA range indicated in the deleted checkpoint record.

System action: The BSDS checkpoint queue is modified as stated. This message is written to the SYSPRINT data set, not to a console.

DSNJ429I RBA RANGE CONFLICTS WITH EXISTING CHECKPOINT RECORD RBA RANGE

Explanation: The specified RBA range for the new checkpoint record either exists or overlaps an existing RBA range in the checkpoint queue.

System action: The current utility is terminated. This message is written to the SYSPRINT data set, not to a console.

User response: Run the print log map (DSNJU004) utility against the SYSUT1 BSDS and SYSUT2 BSDS. Determine the correct RBA range and correct the STARTRBA and ENDRBA parameters and rerun the utility.

DSNJ430I SPECIFIED ENTRY CANNOT BE ADDED WITHOUT OVERLAYING EXISTING LOWEST ENTRY

Explanation: The specified RBA range for the new checkpoint record is less than the existing lowest entry. The checkpoint queue is currently full and cannot add the new entry without overlaying the lowest entry.

System action: The current utility is terminated. This message is written to the SYSPRINT data set, not to a console.

User response: Run the print log map (DSNJU004) utility against the SYSUT1 BSDS and SYSUT2 BSDS. Determine the lowest existing entry and either change the STARTRBA and ENDRBA parameters or delete the existing lowest entry and add the new low checkpoint entry and rerun the utility.

DSNJ431I STARTRBA SPECIFIED NOT FOUND IN CHECKPOINT QUEUE

Explanation: The specified STARTRBA could not be located in the checkpoint queue.

System action: The current utility is terminated. This message is written to the SYSPRINT data set, not to a console.

User response: Run the print log map (DSNJU004) utility against the SYSUT1 BSDS and SYSUT2 BSDS. Determine the correct STARTRBA value and correct the STARTRBA parameter and rerun the utility.

DSNJ432I THE OFFLRBA VALUE MUST END WITH 'FFF'

Explanation: The OFFLRBA value is invalid. It must end with 'FFF'.

System action: The current utility is terminated. This message is written to the SYSPRINT data set, not to a console.

User response: Correct the control statement and rerun the utility.

DSNJ433I *csect-name* **INVALID CONTROL STATEMENT. A MEMBER LIST MUST CONTAIN AT LEAST ONE MEMBER NAME.**

Explanation: The control statement is invalid. The member list is either empty or in the wrong format.

System action: The control statement is ignored. DB2 uses the SYSUT1 DD statement to locate the BSDS to be mapped. If the SYSUT1 DD statement is not available, the utility terminates.

User response: Refer to *DB2 Utility Guide and Reference* for information on creating a print log map (DSNJU004) utility control statement. Correct or remove the control statement and rerun the job.

Problem determination: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Collect the following diagnostic items listed in

Appendix A, "Problem determination," on page 659: 32.

DSNJ434I *csect-name* **SYSIN MUST CONTAIN AT LEAST ONE RECORD.**

Explanation: A SYSIN DD statement was provided, but the data set was null, or the JCL statement "//SYSIN DD *" was immediately followed by the statement "/*".

System action: The error is ignored. DB2 uses the DD SYSUT1 statement to locate the BSDS to be mapped. If the SYSUT1 DD statement is not available, the utility terminates.

User response: Refer to *DB2 Utility Guide and Reference* for information on creating a print log map (DSNJU004) utility control statement. Correct or remove the control statement and rerun the job.

Problem determination: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 32.

DSNJ435I *csect-name* **THE MEMBER LIST OF A CONTROL STATEMENT MUST BE TERMINATED WITH A PARENTHESIS.**

Explanation: The control statement contained the correct format for starting a list of members, but the terminating parenthesis was not found.

System action: The control statement is accepted with an assumed parenthesis after the last member name recognized, and processing continues.

User response: Refer to *DB2 Utility Guide and Reference* for information on creating a print log map (DSNJU004) utility control statement. Correct or remove the control statement and rerun the job.

Problem determination: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 32.

DSNJ436I *csect-name* **THE DD STATEMENT FOR** *ddname* **IS MISSING.**

Explanation: The print log map (DSNJU004) utility requested information about the identified *ddname* from the MVS Dynamic Allocation facility, but the DD statement was not found in the job.

System action: The job is terminated.

User response: Refer to *DB2 Utility Guide and Reference* for information on required JCL statements. Correct the JCL and rerun the job.

Problem determination: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 32.

DSNJ437I *csect-name* **UNABLE TO ALLOCATE BSDS** *dsname***. SVC 99 RTN CODE** *rc***, RSN CODE** *rsncd***, INFO** *infocd***.**

Explanation: The print log map (DSNJU004) utility attempted to allocate a BSDS data set using the MVS Dynamic Allocation facility, but the system was unable to complete the allocation. The return code, reason code, and information code were returned by the system's SVC 99 facility. See *MVS/ESA Programming: Authorized Assembler Services Guide* for an explanation of these codes.

System action: The processing of the BSDS named in the message is terminated. If this is the last (or only) BSDS to be processed by this job, the job is terminated.

User response: Refer to *DB2 Utility Guide and Reference* for information on required JCL statements. Correct the JCL and rerun the job.

Problem determination: If you suspect an error in DB2, refer to Part 2 of *DB2 Utility Guide and Reference* for information on identifying and reporting the problem.

Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 32.

DSNJ438I *csect-name* **THE** *xxx* **VALUE MUST END WITH** *'yyy'*

Explanation: The STARTRBA or ENDRBA value specified on the NEWLOG statement is invalid. *'xxx'* is either STARTRBA or ENDRBA, depending on which parameter is invalid. If *'xxx'* equals STARTRBA, then *'yyy'* equals *'000'* and if *'xxx'* equals ENDRBA, then *'yyy'* equals *'FFF'*.

System action: The current utility is terminated. This message is written to the SYSPRINT data set, not to a console.

User response: Correct the control statement and rerun the utility.

DSNJ439I *csect_name* **NEW FUNCTION MODE REQUIRED FOR** *xxxxxx* **OPERATION**

Explanation: An attempt was made to run a log utility requiring New Function mode while DB2 was in Compatibility mode or Enabling New Function mode.

DB2 must be in New Function mode before the utility

program can run successfully. *xxxxxx* is the name of the attempted operation requiring New Function mode.

System action: The program will terminate with no action taken.

System programmer response: Convert DB2 to New Function mode and retry the program.

DSNJ440I BSDS HAS ALREADY BEEN CONVERTED, DDNAME=*nnnnnnnn*

Explanation: This message occurs when the BSDS conversion program DSNJCNVB detects that the BSDS data set that was specified by DDNAME *nnnnnnnn* has already been converted.

System action: If only SYSUT1 was specified, or both SYSUT1 and SYSUT2 data sets have already been converted, the program will terminate with no action taken. In the case where both are specified, but only SYSUT1 was previously converted, the program will continue to process the conversion for SYSUT2.

System programmer response: No further action is necessary.

DSNJ441I INVALID BSDS CONVERSION, DDNAME=*xxxxxx*

Explanation: This message is issued when a program attempting to access the BSDS data set encounters an invalid BSDS. An invalid BSDS is the result of a failure during a prior attempt to run conversion program DSNJCNVB. *xxxxxx* is the DDNAME of the BSDS data set being accessed.

System action: The program will terminate with no action taken.

System programmer response: The procedure for running program DSNJCNVB includes making a copy of the BSDS prior to conversion. Restore the BSDS to the original pre-conversion copy, and retry the conversion. Once the conversion is successful, retry the program.

DSNJ700I *csect_name* **ERROR ATTEMPTING TO ACCESS BSDS FOR PEER MEMBER ID=** *member-id***, MEMBER NAME=** *member-name***, REASON-CODE=** *reason-code***.**

Explanation: DB2 issues this message to identify the peer member ID, and if known, the peer member name associated with listed DB2 reason code.

System action: An execution unit issues this message and then, if this was an unconditional request, writes a record to SYS1.LOGREC and requests an SVC dump. The execution unit then terminates.

Operator response: Collect all relevant diagnostic materials, including SYS1.LOGREC, SVC dump, and console output of peer and host.

System programmer response: Review the explanation for the associated DB2 reason code for further action.

DSNJ701E *csect-name* **MEMBER NAME (***membername***) IN BSDS DOES NOT MATCH MEMBER NAME (***membername***) IN DB2 SYSTEM PARAMETERS** *dsnzparm*

Explanation: During startup, DB2 detected that the member name listed in its bootstrap data set (BSDS) does not match the member name in the DB2 system parameters load module.

Operator response: Notify the system programmer.

System programmer response: Restore the BSDS or specify a different system parameters load module on the START DB2 command.

DSNJ702E *csect_name* **GENERIC LU NAME** *lu-name* **IN BSDS DOES NOT MATCH THE GENERIC LU NAME** *lu-name* **ASSOCIATED WITH THE DATA SHARING GROUP.**

Explanation: DB2 issues this message during startup when it detects that the generic LU name in the bootstrap data set (BSDS) does not match the generic LU name in the shared communications area (SCA) structure.

System action: DB2 startup continues.

Operator response: Notify the system programmer.

System programmer response: Depending on the accuracy of the BSDS value, take one of the following actions:

- BSDS value is correct: DB2 startup will automatically replace the generic LU name in the SCA with the correct value. The BSDS for each member of this DB2 data sharing group must specify this same value for generic LU name. If any member specifies a different value for generic LU name, use the change log inventory utility (DSNJU003) to change the generic LU name, as described in *DB2 Utility Guide and Reference*.
- BSDS value is not correct: DB2 startup will replace the generic LU name in the SCA. The SCA value will be incorrect when startup is complete. Take the following actions to correct this situation:
  - Stop this DB2 subsystem.
  - Use the change log inventory utility (DSNJU003) to change the generic LU name of this member, as described in *DB2 Utility Guide and Reference*.
  - Issue START DB2 for this member. START DB2 will correct the generic LU name contained in the SCA.

DSNJ703E *csect-name* **GROUP NAME (***groupname***) IN BSDS DOES NOT MATCH GROUP NAME (***groupname***) IN DB2 SYSTEM PARAMETERS** *dsnzparm*

Explanation: During startup, DB2 detected that the group name listed in its bootstrap data set (BSDS) does not match the group name in the DB2 system parameters load module.

Operator response: Notify the system programmer.

System programmer response: Restore the BSDS or specify a different system parameters load module on the START DB2 command.

DSNJ704I *csect-name* **NEW MEMBER(***member-name***) START-UP NOT ALLOWED WHEN GROUP-RESTART IS REQUIRED FOR GROUP(***group-name***).**

Explanation: One of the following occurred:

- A new member *member-name* tried to start when a group restart of data sharing group *group-name* is required.
- A member other than the originating member of a data sharing group attempted to start the new DB2 data sharing group. The originating member of a group is either an existing DB2 subsystem which formed the data sharing group or a new DB2 subsystem which was installed to form the group.

System action: The DB2 subsystem stops.

Operator response: Start a previously active member of this group. When the required group restart is completed, start the new member again.

If this is the very first startup of the group, you must start the originating member first before starting any other member.

DSNJ705I *csect-name* **THE DB2 SYSTEM PARAMETERS SPECIFY DATA SHARING, BUT THE BSDS SPECIFIES NON-DATA SHARING.**

Explanation: The system parameters passed on the DB2 start command specified data sharing, but the bootstrap data set (BSDS) did not indicate this DB2 is a data sharing member.

System action: The DB2 subsystem stops.

Operator response: If this DB2 should be a member of a data sharing group, inform the system programmer. If this DB2 is not a member of a data sharing group, restart DB2 with the correct system parameters load module

System programmer response: If this DB2 should be a member of a data sharing group, execute the installation job DSNTIJUZ, which updates the BSDS for data sharing.

If this DB2 is not a member of a data sharing group, reassemble the system parameters load module and specify DSHARE=NO.

If this DB2 is attempting to disable the data sharing group, then reassemble the system parameters load module and specify DSHARE=NO, then execute the installation job DSNTIJGF, and then restart this DB2. Job DSNTIJGF must be used with caution, this will disable the data sharing group. For more information, see Chapter 2 of *DB2 Data Sharing: Planning and Administration*.

DSNJ706I *csect-name* **THE DB2 SYSTEM PARAMETERS SPECIFY NON-DATA SHARING, BUT THE BSDS SPECIFIES DATA SHARING.**

Explanation: The system parameters passed on the DB2 START command specified non-data sharing, but the bootstrap data set (BSDS) indicates this is a data sharing member.

System action: The DB2 subsystem stops.

Operator response: If this DB2 is a member of a data sharing group, restart DB2 with the correct system parameters load module. If this DB2 is not a member of a data sharing group, inform the system programmer.

System programmer response: If this DB2 should be a member of a data sharing group, reassemble the system parameters load module and specify DSHARE=YES. If this DB2 is not a member of a data sharing group, execute the installation job DSNTIJGF, which updates the BSDS for non-data sharing.

DSNJ707E *csect_name* **LOCATION NAME** *location-name* **IN BSDS DOES NOT MATCH THE LOCATION NAME** *location-name* **ASSOCIATED WITH THE DATA SHARING GROUP.**

Explanation: DB2 issues this message during startup when it detects that the LOCATION name in the bootstrap data set (BSDS) does not match the LOCATION name in the system communications area (SCA) structure.

System action: DB2 startup continues.

Operator response: Notify the system programmer.

System programmer response: Depending on the accuracy of the BSDS value, take one of the following actions:

- BSDS value is correct: DB2 startup will automatically replace the LOCATION name in the SCA with the correct value. The BSDS for each member of this DB2 data sharing group must specify this same value for LOCATION. If any member specifies a different value for LOCATION, use the change log inventory utility (DSNJU003) to change the LOCATION name, as described in *DB2 Utility Guide and Reference* .

- BSDS value is not correct: DB2 startup will replace the LOCATION name in the SCA. The SCA value will be incorrect when startup is complete. Take the following actions to correct this situation:
  - Stop this DB2 subsystem.
  - Use the change log inventory utility (DSNJU003) to change the LOCATION name of this member, as described in *DB2 Utility Guide and Reference*.
  - Issue START DB2 for this member. START DB2 will correct the LOCATION name contained in the SCA.

DSNJ708I *csect-name* **NEW MEMBER START-UP IS NOT ALLOWED WHEN THE DATA SHARING GROUP IS IN SYSTEM RECOVER PENDING MODE**

Explanation: A new data sharing member attempted to join the data sharing group when the data sharing group is in System Recover Pending mode.

System action: DB2 startup terminates.

Operator response: When the data sharing group is no longer in System Recover Pending mode, start the new member again.

DSNJ711E *csect-name* **SCA OPEN FAILURE ENCOUNTERED.**

Explanation: A failure occurred during command processing in an attempt to open an object in the SCA.

System action: Command processing terminates unsuccessfully.

System programmer response: Obtain SYS1.LOGREC and possible SVC dump for analysis.

Problem determination: Review the DB2 system console for possible DSN7xxxx and/or SVC dump messages that could relate to this problem.

If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNJ712E *csect-name* **SCA READ FAILURE ENCOUNTERED.**

Explanation: A failure occurred during command processing attempting to read from the SCA.

System action: Command processing terminates unsuccessfully.

System programmer response: Obtain SYS1.LOGREC and possible SVC dump for analysis.

Problem determination: Review the DB2 system console for possible DSN7xxxx and/or SVC dump messages that could relate to this problem.

If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNJ713E *csect-name* **SCA WRITE FAILURE ENCOUNTERED.**

Explanation: A failure occurred during command processing attempting to write to the SCA.

System action: Command processing terminates unsuccessfully.

System programmer response: Obtain SYS1.LOGREC and possible SVC dump for analysis.

Problem determination: Review the DB2 system console for possible DSN7xxxx and/or SVC dump messages that could relate to this problem.

If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNJ990I *csect-name* **SHOWCB ERROR, RETURN CODE =** *error-code*

Explanation: SHOWCB failed with error-code *error-code*. An error-code of 4 means that the log data set was allocated greater than 4 gigabytes in size, which is not supported.

Severity: 8 (error).

System action: Processing terminates.

User response: Notify the system programmer.

System programmer response: Follow standard VSAM diagnostic procedure. Refer to *z/OS DFSMS: Macro Instructions for Data Sets*. If the log data set was allocated greater than 4 gigabytes in size, reallocate the data set less than 4 gigabytes in size.

DSNJ991I *csect-name* **START OF LOG DATA SET PREFORMAT FOR JOB** *job-name step-name*

Explanation: The active log data set preformat utility has started.

Severity: 0 (informational)

DSNJ992I *csect-name* **LOG DATA SET NAME =** *data-set-name*

Explanation: The indicated *data-set-name* is to be preformatted.

Severity: 0 (informational)

DSNJ993I *csect-name* **LOG DATA SET IS NOT A VSAM DATA SET**

Explanation: The input log dataset is not a VSAM data set.

Severity: 8 (error)

System action: Processing terminates.

User response: Use the AMS IDCAMS Service to define the active log data set as a VSAM Linear data set.

DSNJ994I *csect-name* **VSAM OPEN ERROR, ACBERRFLG =** *error-code*

Explanation: VSAM OPEN failed with the indicated ACB *error-code*.

Severity: 8 (error) if error is greater than or equal to 128; otherwise, 4 (warning)

System action: Processing terminates if the error code is greater than or equal to 128; otherwise, processing continues normally.

User response: Notify the system programmer.

System programmer response: Follow standard VSAM diagnostic procedure. Refer to *z/OS DFSMS: Macro Instructions for Data Sets*. See also *MVS/DFP V3R3 Diagnosis Guide* for more information about the VSAM procedures.

DSNJ995I *csect-name* **VSAM PUT ERROR,RPLERREG =** *return-code*, **RPLERRCD =** *reason-code*

Explanation: VSAM PUT failed with the indicated *return-code* and *reason-code*.

Severity: 8 (error)

System action: Processing terminates.

User response: Notify system programmer.

System programmer response: Follow standard VSAM diagnostic procedure. Refer to *z/OS DFSMS: Macro Instructions for Data Sets*. See also *MVS/DFP V3R3 Diagnosis Guide* for more information about the VSAM procedures.

DSNJ996I *csect-name* **LOG PREFORMAT COMPLETED SUCCESSFULLY,nn RECORDS FORMATTED**

Explanation: The active log data set has been preformatted successfully. 'nn' records (or VSAM control intervals) have been formatted.

Severity: 0 (informational)

DSNJ997I *csect-name* **LOG PREFORMAT TERMINATED**

Explanation: Preformatting of the active log data set did not complete successfully.

Severity: 8 (error)

System action: Processing terminates.

User response: Examine other error messages associated with this job.

DSNJ998I *csect-name* **LOG DATA SET IS NOT EMPTY**

Explanation: The input log data set is not an empty data set. The input data set must be empty to prevent the loss of actual log data.

Severity: 8 (error)

System action: Processing terminates.

User response: Use the AMS IDCAMS Service to delete and redefine the active log data set or correct the input log data set name.

Chapter 10. DSNL... messages

DSNL001I DDF IS ALREADY STARTED

Explanation: The distributed data facility (DDF) was already active when you issued -START DDF command.

System action: The issued command is not executed.

DSNL002I DDF IS ALREADY STOPPED

Explanation: The distributed data facility (DDF) had already been stopped when the -STOP DDF command was issued.

System action: The issued command is not executed.

DSNL003I DDF IS STARTING

Explanation: The distributed data facility (DDF) -START command is starting.

System action: DB2 begins processing the -START DDF command.

DSNL004I *csect* **DDF START COMPLETE LOCATION** *locname* **LU** *netname.luname* **GENERICLU** *netname.gluname* **DOMAIN** *domain* **TCPPORT** *tcpport* **RESPORT** *resport*

Explanation: The distributed data facility (DDF) started.

locname The LOCATION of the DB2 subsystem.

netname.luname
The fully qualified LU name for the DB2 subsystem.

netname.gluname
The fully qualified VTAM generic LU name for the DB2 subsystem. The value '-NONE' indicates that no VTAM generic LU name is used by the DB2 subsystem.

domain For non-data sharing systems, domain is the name returned by the DNS for the system's default IP address. For data sharing systems, domain is the name that represents the DB2 group. For the remote DB2 data sharing group, the DRDA requester should use a domain name of the following form:

```
location.sysplex.domainname
```

where *location* is the DB2 location name, *sysplex* is the sysplex name in the COUPLExx parmlib member and *domainname* is the domain name specified in the TCPIP.DATA file. If you are an MVS requester, the domain name is specified in column IPADDR of catalog table SYSIBM.IPNAMES. If you are a DB2 Connect requester, the CATALOG TCPIP NODE command (or the equivalent command for your environment) is used to specify the TCP/IP domain name. The value '-NONE' indicates that TCP/IP is not supported, TCP/IP initialization failed, or DB2 has been started in restricted access mode.

tcpport The TCP/IP port number used by this DB2 subsystem to accept incoming DRDA connection requests.

- A zero indicates TCP/IP is not supported. The BSDS DDF record is not configured with a PORT.
- A value of '-NONE' indicates that DB2 will not accept connections on this port.

  This may be due to specifying 0 for the DSN6SYSP MAXDBAT installation parameter defined in DSNZPARM. For more information, see the "MAX REMOTE ACTIVE" specification of the DSNTIPE installation panel in the *DB2 Installation Guide*.

resport The TCP/IP port number used by this DB2 subsystem to accept incoming two-phase commit resynchronization requests.

A zero indicates TCP/IP is not supported. The BSDS DDF record is not configured with a RESPORT.

System action: The DDF start up process completed. However, it might fail for reasons specified by messages DSNL013I, DSNL015I, DSNL016I, DSNL017I, DSNL018I, DSNL019I, DSNL700I, or DSNL705I.

System programmer response: Use the recommended domain name when configuring network definitions for TCP/IP requesters.

DSNL005I DDF IS STOPPING

Explanation: The distributed data facility (DDF) -STOP command has been initiated.

System action: The -STOP DDF command is initiated.

DSNL006I DDF STOP COMPLETE

Explanation: The distributed data facility (DDF) -STOP command has completed.

System action: The DDF has stopped.

© Copyright IBM Corp. 1983, 2008

DSNL007I DDF IS ABNORMALLY TERMINATING

Explanation: Abnormal termination of the distributed data facility (DDF) has started.

System action: Abnormal termination of the DDF is starting.

DSNL008I DDF ABNORMAL TERMINATION COMPLETE

Explanation: Abnormal termination of the distributed data facility (DDF) has completed.

System action: The DDF has stopped.

DSNL009I INVALID DDF THREAD *logical-unit-of-work-id* **SPECIFIED**

Explanation: No distributed transaction with the specified 'logical-unit-of-work-id' exists.

System action: The specified thread is not canceled.

User response: Determine the correct 'logical-unit-of-work-id' of the distributed thread that is to be canceled and then reissue the command.

Problem determination: Determine the correct value. The -DISPLAY THREAD command can be used to display all distributed activity.

DSNL010I DDF THREAD *logical-unit-of-work-id* **HAS BEEN CANCELED**

Explanation: The distributed data facility (DDF) thread 'logical-unit-of-work-id' was canceled.

If 'logical-unit-of-work-id' is an LUWID (LU network name and an LUW instance number) instead of an LUWID TOKEN, then in rare circumstances more than one DDF thread might be canceled.

This situation only arises when an application accesses the same DB2 server by different access paths (for example, A->B->C and A->D->C). In such a case, multiple threads with the same LUWID exist at the server (for example, C). If CANCEL DDF THREAD is issued at the server specifying this LUWID, every thread at the server with the LUWID is canceled.

System action: The specified thread is canceled.

DSNL011I VTAM STOPPING DDF

Explanation: The operator issued a HALT NET command to terminate VTAM.

System action: The distributed data facility (DDF) -STOP command is initiated.

DSNL012I VTAM TERMINATING DDF

Explanation: The operator issued a HALT NET QUICK command or a HALT NET CANCEL command, or VTAM is terminating abnormally. This message can be issued any time after -START DDF and occurs when VTAM is stopped, which can occur prior to DDF termination. This message is not issued when DDF terminates normally.

System action: Termination of the distributed data facility is initiated.

DSNL013I VTAM OPEN ACB FAILED ERROR=*error-code*

Explanation: VTAM returned a nonzero return code after DB2 invoked the OPEN ACB macro instruction during termination of the distributed data facility (DDF). VTAM return codes for OPEN error conditions are defined in *z/OS Communications Server SNA Programming*.

System action: Termination of the distributed data facility is initiated.

Operator response: Notify the system programmer.

System programmer response: Determine the cause of the OPEN ACB failure, correct the error, and issue the -START DDF command.

Problem determination: Refer to *z/OS Communications Server SNA Programming* for failure analysis during an OPEN macro instruction.

DSNL014I VTAM CLOSE ACB FAILED ERROR=*error-code*

Explanation: VTAM returned a nonzero return code after DB2 invoked the CLOSE ACB macro instruction during termination of the distributed data facility (DDF). VTAM return codes for CLOSE error conditions are defined in *z/OS Communications Server SNA Programming*.

System action: Termination of DDF continues.

Operator response: Notify the system programmer.

System programmer response: Determine the cause of the failure, fix the error condition, and then issue the -START DDF command.

Problem determination: Refer to *z/OS Communications Server SNA Programming* for failure analysis during a CLOSE macro instruction.

DSNL015I VTAM SETLOGON FAILED RTNCD=*error-code* **FDBK2=***open-flags*

Explanation: VTAM returned a nonzero return code after DB2 invoked the SETLOGON macro during initialization of the distributed data function (DDF).

VTAM return codes are defined in *z/OS Communications Server SNA Programming*.

System action: Termination of the DDF is started.

Operator response: Notify the system programmer.

System programmer response: Refer to *z/OS Communications Server SNA Programming* to determine the cause of the failure. After the error is fixed, issue the -START DDF command.

DSNL016I DDF RECORD NOT FOUND IN BSDS

Explanation: DDF cannot find the DDF communication record in the BSDS.

System action: DDF termination process is initiated.

Operator response: Notify the system programmer.

System programmer response: Use the change log inventory to add the DDF communication record to the BSDS.

Problem determination: Collect the following diagnostic item:

- Print the contents of the BSDS using the print log map utility.

DSNL017I INVALID DDF RECORD FOUND IN BSDS

Explanation: A invalid or incomplete DDF communication record was found in the BSDS.

System action: DDF termination process is initiated.

Operator response: Notify the system programmer.

System programmer response: Use the change log inventory to add a valid DDF communication record to the BSDS.

Problem determination: Collect the following diagnostic item:

- Print the contents of the BSDS using the print log map utility.

DSNL018I VSAM ERROR READING DDF RECORD IN BSDS RPLERRCD=*error-code*

Explanation: A VSAM GET command was issued that resulted in a nonzero return code. RPLERRCD 'error-code' is the VSAM return code.

System action: Termination of the distributed data facility (DDF) is initiated.

Operator response: Notify the system programmer.

System programmer response: Determine the cause of the failure, fix the error condition, and issue the -START DDF command.

The cause of the failure is dictated by the VSAM return code. The BSDS may have to be recovered by using a backup copy.

Problem determination: Refer to the appropriate VSAM publications.

DSNL019I VTAM DOES NOT SUPPORT APPC FUNCTIONS

Explanation: The installed release of VTAM does not support the advanced program-to-program communications (APPC) application program interface (API). The APPC API bit is not set in the VTAM function-list vector in the access-method-support vector list.

System action: Termination of the distributed data facility (DDF) is initiated.

Operator response: Notify the system programmer.

System programmer response: Determine whether the installed release of VTAM supports the APPC API. If VTAM does not support the APPC API, a VTAM release that supports the APPC API must be installed before the DDF can be started.

DSNL020I KEYWORD 'THREAD' PARAMETER *parameter* **FORMAT IN ERROR**

Explanation: The parameter value specified for the logical-unit-of-work ID on the CANCEL DDF THREAD has an invalid format. The correct form is 'netid.luname.luwunique', where:

netid A 1- to 8-character network ID.

luname
A 1- to 8-character luname.

luwunique
A 12-character hex number.

Luname and luwunique are separated by a '.' delimiter. Netid is optional. If netid is specified, then netid and luname are separated by a '.' delimiter. If netid is not specified, then the '.' delimiter before luname is not present.

System action: The specified thread is not canceled.

User response: Determine the correct logical-unit-of-work ID of the distributed thread that is to be canceled and reissue the command.

Problem determination: Determine the correct value. The -DISPLAY THREAD command can be used to display all distributed activity.

DSNL021I *command* **ACCEPTED**

Explanation: The 'command' command has been accepted and asynchronous processing of 'command' has been initiated. The command can only be -START or -STOP the distributed data facility.

System action: The command was accepted and asynchronous processing has been initiated.

DSNL022I *csect-name* **DDF THREAD** *logical-unit-of-work-id* **WAS PREVIOUSLY CANCELED**

Explanation: A command to cancel the distributed data facility (DDF) thread *logical-unit-of-work-id* was previously issued. Either the CANCEL DDF THREAD or the CANCEL THREAD command was entered. The thread is currently being canceled. The CANCEL command terminates a thread only if the thread is processing in DB2 or is currently suspended by DB2. The thread might not be processing in DB2 in these situations:

- The thread is processing in the application and has not made a request to DB2. The DISPLAY THREAD command indicates this.
- The user placed the thread on an MVS resource using the ENQUE command. The CANCEL command has no effect on the thread until ENQUE completes and the thread returns to DB2.
- The thread is processing in VTAM and is waiting for a response from a partner. The DISPLAY THREAD DETAIL command indicates this. In this case the thread does not terminate until VTAM returns to DB2. VTAM commands, which will terminate the session or LU, can be used to cause VTAM to return to DB2.

DSNL023I DDF IS ALREADY IN THE PROCESS OF STARTING

Explanation: The distributed data facility (DDF) was starting when the -START DDF or -STOP DDF command was issued.

System action: The issued command is not executed.

DSNL024I DDF IS ALREADY IN THE PROCESS OF STOPPING

Explanation: The distributed data facility (DDF) was stopping when the -STOP DDF or -START DDF command was issued.

System action: The -START or -STOP DDF command is not executed.

DSNL025I A DISTRIBUTED AGENT FOR *luw-id=token* **EXISTS AT THIS LOCATION BUT NOT AT SITE WITH LOCATION** *location* **AND LU** *luname*

Explanation: The DDF error monitor has detected that a distributed agent exists at this location but not at the named partner location.

Where

luw-id=token
The logical unit of work ID of the thread followed by an '=' sign and a token which may be used in place of 'luw-id' in any DB2 command that accepts 'luw-id' as input.

location
The location name of the partner site where a thread with the same 'luw-id' does not exist.

luname
The 8-character LUNAME of the partner location.

Operator response: Determine if the inconsistency is due to network conditions or a DB2 error. If a DB2 error is suspected, acquire documentation to diagnose the problem.

1. Console dump of DB2 address spaces (*ssnm*DBM1, *ssnm*MSTR, *ssnm*DIST), VTAM, and user's address space at the requesting site.
2. Console dump of DB2 address spaces (*ssnm*DBM1, *ssnm*MSTR, *ssnm*DIST), and VTAM at the server site.
3. SYSLOG from requesting and server sites.
4. LOGREC from requesting and server sites.

If an indefinite inconsistency is suspected, use the CANCEL DDF THREAD command to terminate the thread. If CANCEL DDF THREAD fails to terminate the agent, use VTAM commands to terminate the thread conversations.

Problem determination: Use the DB2 DISPLAY THREAD command with DETAIL option at each DB2 subsystem to display information about the agent and its conversations. Use VTAM commands to display additional information about the conversations being used by the thread and also about network conditions.

Collect the following diagnostic items listed in Appendix A, “Problem determination,” on page 659: 1, 3, 5.

DSNL026I LUW-ID *luw-id=token* **APPEARS TO BE SUSPENDED PARTNER LOCATION** *location* **PARTNER LU** *luname*

Explanation: The DDF error monitor has detected that a distributed agent with the specified 'luw-id' at this location and at the named partner 'location' may be suspended indefinitely.

Where

luw-id=token
The logical unit of work ID of the thread followed by an '=' sign and a token which may be used in place of 'luw-id' in any DB2 command that accepts 'luw-id' as input.

location
The location name of the partner site where a thread with the same 'luw-id' exists.

luname
The 8-character LUNAME of the partner location.

Operator response: Determine if the inconsistency is due to network conditions or a DB2 error. If a DB2 error is suspected, acquire documentation to diagnose the problem.

1. Console dump of DB2 address spaces (*ssnm*DBM1, *ssnm*MSTR, *ssnm*DIST), VTAM, and hung user's address space at the requesting site.
2. Console dump of DB2 address spaces (*ssnm*DBM1, *ssnm*MSTR, *ssnm*DIST), and VTAM as the server site.
3. SYSLOG from requesting and server sites.
4. LOGREC from requesting and server sites.

If an indefinite inconsistency is suspected, use the CANCEL DDF THREAD command to terminate the thread. If CANCEL DDF THREAD fails to terminate the agent, use VTAM commands to terminate the thread conversations.

Problem determination: Use the DB2 DISPLAY THREAD command with DETAIL option at each DB2 subsystem to display information about the agent and its conversations. Use VTAM commands to display additional information about the conversations being used by the thread and also about network conditions.

Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 3, 5.

DSNL027I *dis-agent-type* **DISTRIBUTED AGENT WITH LUW-ID** *luw-id=token* **THREAD-INFO** *thread-info* **RECEIVED ABEND=***abnd* **REASON=***reason*

Explanation: A distributed agent has abended at this location. This message will be followed by zero or more occurrences of DSNL028I, each of which describes a remote location where the distributed agent also exists and where diagnostic information might also need to be collected. This message is informational and with the DSNL028I message(s) identifies the remote locations where diagnostic information may have to be collected.

Where

dis-agent-type
'REQUESTING' if the thread that abended is an allied thread that has distributed activity and SERVER if the thread that abended is a database access thread.

luw-id=token
The logical-unit-of-work ID of the thread followed by an '=' sign and a 5 digit token value which may be used in any DB2 command that accepts luw-id as input.

thread-info
Thread-info contains additional identifiers that define more specific information regarding the identity of the thread.

In many distributed configurations, the primary authorization-ID used with DB2 is not necessarily the user-ID of the end user.

The following identifiers, separated by a colon(:), are provided:

- The primary authorization-ID associated with the thread.
- Work station name of the end user's work station. An asterisk (*) indicates that it is not relevant to the thread or was not provided by the client.
- End user's user-ID. An asterisk (*) indicates that it is not relevant to the thread or was not provided by the client.
- Application name of the end user's application. An asterisk (*) indicates that it is not relevant to the thread or was not provided by the client.

abnd The abend code or zeros if the abend code is indeterminate.

reason The DB2 reason code or zeros if the reason code is indeterminate.

Problem determination: Gather appropriate diagnostic information (refer to the System Programmer Response section of the abend reason code specified in this message if applicable). Refer to DSNL028I message(s) for the names of the other location(s).

DSNL028I *luw-id=token* **ACCESSING DATA** *preposition* **LOCATION** *location nettype netvalue*

Explanation: This message follows DSNL027I and identifies a remote location where work might have been occurring for the distributed thread. This message is informational and with the DSNL027I message identifies a remote location where diagnostic information might have to be collected.

luw-id=token
The logical unit-of-work ID of the thread followed by an '=' sign and a token which may be used in place of 'luw-id' in any DB2 command that accepts 'luw-id' as input.

preposition
'FOR' if the thread is a database access thread and 'AT' if the thread is an allied thread that has distributed activity.

location
The LOCATION name of the remote site.

nettype
Identifies the type of network used to access the remote site. There are two possible values:

LU -- indicates that VTAM is used to connect to the remote site.

IPADDR --
indicates that TCP/IP is used to connect to the remote site.

netvalue
The network address of the remote location. *netvalue* can be blank.

- If *nettype* is LU, *netvalue* specifies the VTAM LU name of the remote location.
- If *nettype* is IPADDR, *netvalue* specifies the dotted decimal IP address of the remote location.

Problem determination: See message DSNL027I.

DSNL029I DDF IS NOT INITIALIZED

Explanation: A -START DDF or -STOP DDF command was issued before DDF was initialized during DB2 startup processing.

Operator response: Wait for the DSN9022I 'START DB2' NORMAL COMPLETION message and re-issue the command.

DSNL030I *csect-name* **DDF PROCESSING FAILURE FOR LUWID=***luwid* **AUTHID=***authid* **REASON=***reason*

Explanation: A requesting conversation was terminated because of DDF processing failure. The requesting agent is identified by the 'luwid' and 'authid' values, if they are available. If the authorization ID is not available, the value '-UNKNOWN' will be displayed. If the 'luwid' is not available, the string 'netid.luname.000000000000' will be displayed, describing the LU that made the request.

The reason code in this message describes the nature of the DDF processing failure.

System action: The system will eventually abend the database access agent.

Operator response: Notify the system programmer for analysis.

System programmer response: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: Inspect the reason code and determine if the system is operating as expected. Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 5.

DSNL031I *csect-name* **DRDA EXCEPTION CONDITION IN RESPONSE FROM SERVER LOCATION=***location* **FOR THREAD WITH LUWID=***luwid* **REASON=***reason code* **ERROR ID=***module_name and source id* **CORRELATION ID=***correlation-id* **CONNECTION ID=***connection-id* **IFCID=***ifcid* **SEE TRACE RECORD WITH IFCID SEQUENCE NUMBER=***seq-no*

Explanation: A DRDA exception condition was detected in response data returned from the application server (AS) with location name *location*.

A DRDA exception condition, detected at the application requester, is one of the following:

- An error reply message and severity code received from the server.
- A parsing error detected while processing a reply message or reply data object
- A DRDA protocol violation, (for example, an object returned that is not valid for this command).

The variables have the following meanings:

luwid The logical unit of work (LUW) ID that identifies the thread and is also included in the trace record.

reason The DB2 reason code for this specific error.

error id
The module name concatenated with a 4-byte source id that uniquely determines the source within the module where the exception is encountered.

correlation-id
The correlation identifier of the thread that detected the DRDA exception condition. Correlation-IDs beginning with '0nn', where nn is a number from 01 to 28, identify system agents.

connection-id
The connection name of the thread that detected the DRDA exception condition.

ifcid The IFCID number of the trace record that was written.

seq-no The IFCID sequence number associated with the trace record. The trace record contains information that can be used to determine what caused the DRDA exception condition, and is written to statistics class 4. Statistics class 4 is directed to SMF by default during DB2 installation, and must be directed to a permanent destination (SMF or GTF) for the trace record to be accessible. The message and the trace record are only written the first time

an error is detected. This avoids flooding the console and the trace. The sequence number is printed in hexadecimal.

If the sequence number is zero this has been caused by one of two conditions:

- statistics class 4 was not directed to a permanent destination
- statistics class 4 was directed to the resident trace table as its permanant destination.

If this message first occurred because statistics class 4 was not directed to a permanent destination, then the same message recurs later (along with trace record) when the statistics class is directed to the resident table as its permanent destination instead of SMF or GTF.

System action: An alert was generated. Additionally, the system wrote a trace record with IFCID=*ifcid* to statistics class 4 that can be used to diagnose the problem.

Problem determination: Determine the destination of statistics class 4 and locate and print the trace record with SEQUENCE NUMBER=*seq-no*. The reason code is described in *DB2 Codes*. The description of the reason code can be used in conjunction with the publication Distributed Relational Database Architecture Reference to determine the type of DRDA exception that has been detected. Use the procedures described in Part 4 of *DB2 Diagnosis Guide and Reference* to determine the problem and which system is responsible.

DSNL032I *csect-name* **DRDA EXCEPTION CONDITION IN REQUEST FROM REQUESTER LOCATION=***location* **FOR THREAD WITH LUWID=***luwid* **REASON=***reason code* **ERROR ID=***module_name and source id* **IFCID=***ifcid* **SEE TRACE RECORD WITH IFCID SEQUENCE NUMBER=***seq-no*

Explanation: A DRDA exception condition was detected while processing a command or command data received from the application requester (AR) with location name *location*. A DRDA exception condition, detected at the application server (AS), is a parsing error detected while processing a command or a command data object.

location
This field contains one of the following:

- The requester's relational database name (location name)
- The requester's VTAM LU name (<luname>)
- The requester's dotted decimal IP address (nnn.nnn.nnn.nnn)

luwid The logical unit of work (LUW) ID that identifies the thread and is also included in the trace record, or 'NETID.LUNAME.000000000000' if the error detected involves an invalid FMH5.

reason The DB2 reason code for this specific error.

error id
The module name concatenated with a 4-byte source id that uniquely determines the source within the module where the exception was encountered.

ifcid The IFCID sequence number associated with the trace record. The trace record contains information that can be used to determine what caused the DRDA exception condition, and is written to statistics class 4. Statistics class 4 is directed to SMF by default during DB2 installation, and must be directed to a permanent destination (SMF or GTF) for the trace record to be accessible. The message and the trace record are only written the first time an error is detected. This avoids flooding the console and the trace. The sequence number is printed in hexadecimal.

If the sequence number is zero this has been caused by one of two conditions:

- statistics class 4 was not directed to a permanent destination
- statistics class 4 was directed to the resident trace table as its permanant destination.

If this message first occurred because statistics class 4 was not directed to a permanent destination, then the same message recurs later (along with trace record) when the statistics class is directed to the resident table as its permanent destination instead of SMF or GTF.

System action: An alert was generated and the system wrote a trace record with IFCID=*ifcid* to statistics class 4 that can be used to diagnose the problem.

Operator response: Notify the system programmer for analysis.

System programmer response: Determine the destination of statistics class 4 and locate and print the trace record with SEQUENCE NUMBER=*seq-no*. The reason code is described in *DB2 Codes*. The description of the reason code can be used in conjunction with the publication Distributed Relational Database Architecture Reference to determine the type of DRDA exception that has been detected. Use the procedures described in *DB2 Diagnosis Guide and Reference* to determine the problem and which system is responsible.

DSNL033I *csect-name* **DDF TERMINATION SCHEDULED DUE TO DETECTION OF ABEND** *abend* **REASON** *reason*

Explanation: A severe error was detected during DDF processing. DDF processing can no longer continue so

DDF termination has been scheduled. 'Csect-name' is the DB2 module that detected the condition and requested DDF termination.

The abend and reason might also contain additional information about the cause.

abend The abend code or zeros if no abend is associated with the termination of DDF.

If the abend was a DB2 04E abend, the DB2 reason code will contain additional information about the abend.

reason The DB2 reason code or zeros if no DB2 reason code is associated with the termination of DDF.

If no abend is associated with the problem, the reason code describes the cause for the termination.

System action: Termination of the distributed data facility (DDF) is initiated.

Operator response: Notify the system programmer for analysis.

System programmer response: Determine the cause of the failure. Correct the condition and restart DDF using the -START DDF command.

Problem determination: If an abend is associated with the failure, a dump might have been provided. LOGREC will also contain information about the abend.

DSNL034E DDF CANNOT BE STARTED BECAUSE OF BSDS INCONSISTENCIES, ORIGINAL LUNAME=*luname1*, BSDS LUNAME=*luname2*, ORIGINAL LOCATION=*location1*, BSDS LOCATION=*location2*

Explanation: The LUNAME or the LOCATION name stored in the bootstrap data set (BSDS) was changed since the distributed data facility (DDF) was last started, but DDF still has resynchronization responsibility with one or more partner logical units using the old LUNAME and LOCATION name. Both the original and current BSDS values of LUNAME and LOCATION name are displayed.

luname1
The LUNAME required for resynchronization

luname2
The LUNAME currently stored in the BSDS

location1
The LOCATION name required for resynchronization

location2
The LOCATION name currently stored in the BSDS

System action: DDF cannot be started until the inconsistencies are corrected.

Operator response: Inform the database administrator.

Database Administrator Action: You have the following options, depending on whether there is a requirement to change the LUNAME or LOCATION name in the BSDS:

- If the names were changed in error, use the change log inventory (DSNJU003) utility to change the LUNAME and/or LOCATION name back to the correct value. Then START DDF.
- If the name must be changed, you must first remove all of the resynchronization responsibility that DB2 had using the old names. You can purge this information, allow automatic resynchronization to take place, or use a combination of both.

  You can purge resynchronization information by using the RESET INDOUBT command with the FORCE option. You can purge all information regardless of partner name (LOCATION (*)), or you can selectively purge by specifying a location name. DDF does not have to be started to use this command. You can use the DISPLAY THREAD command to generate a report of indoubt threads. This report indicates all resynchronization responsibility known by DB2.

  If you do not purge the resynchronization information, or only selectively purge, and you want automatic resynchronization to occur before changing the names, you must first use the change log inventory (DSNJU003) utility to change the LUNAME and/or LOCATION name back to the correct value. Then START DDF. At this point, automatic resynchronization begins. You can monitor the progress by periodically using the DISPLAY THREAD command for TYPE (INDOUBT) threads. Resynchronization does not complete until all needed communications paths are available and all needed partner logical units are available. If you cannot wait for availability, use the RESET INDOUBT command with the FORCE option to purge information associated with unavailable partners.

DSNL035I DDF CANNOT BE STOPPED BECAUSE AN INDOUBT DATA BASE ACCESS THREAD EXISTS

Explanation: DDF is the connection manager for a database access thread (DBAT) that is indoubt. A STOP DDF MODE(QUIESCE) command is rejected until the indoubt logical unit of work is resolved.

System action: DDF is not stopped.

Operator response: Inform the database administrator.

Database Administrator Action: You have the following options:

- Wait for automatic resynchronization to be performed. When this occurs, either message DSNL407I or DSNL415I is displayed.

  After the indoubt thread is resolved, reissue the STOP DDF command.
- Use the DISPLAY THREAD TYPE(INDOUBT) command to find the LOCATION that has coordinator responsibility for the indoubt LUW. Contact that location and determine whether the LUW committed or aborted. Then use the RECOVER INDOUBT command to cause the indoubt DBAT to be resolved with the same decision.

  After the indoubt thread is resolved, reissue the STOP DDF command.
- Issue the STOP DDF MODE(FORCE) command. The indoubt thread remains indoubt and can be resolved by using automatic resynchronization after DDF is started again, or it can be resolved manually as described above.

DSNL036I **DDF THREAD** *logical-unit-of-work-id* **SPECIFIED IS INDOUBT**

Explanation: A database access thread that is indoubt cannot be canceled by using the -CANCEL DDF THREAD command.

System action: The specified thread is not canceled.

Operator response: Use the -DISPLAY THREAD TYPE(*) command to display all distributed activity.

DSNL037I *csect-name* **RESET GENERICLU** *lu* **SUCCESSFUL**

Explanation: The RESET GENERICLU command successfully purged the VTAM sysplex information from shared memory.

lu The value supplied on the RESET GENERICLU command

System action: The VTAM information contained in the sysplex shared memory for the indicated LU is purged.

DSNL038E *csect-name* **RESET GENERICLU** *lu* **FAILED DUE TO VTAM RTNCD=***xx* **and FDB2=***yy*

Explanation: The VTAM CHANGE,OPTCD=(SYN,ENDAFFIN) macro returned an unexpected return code, which is described by the indicated VTAM RTNCD and FDB2 values.

lu The value supplied on the RESET GENERICLU command

xx The hexadecimal value of the VTAM RTNCD field

yy The hexadecimal value of the VTAM FDB2 field

System action: VTAM was not able to purge the sysplex shared memory information for the indicated LU.

System programmer response: Refer to *z/OS Communications Server SNA Programming* for the meaning of the VTAM RTNCD and FDB2 values.

DSNL039E *csect-name* **RESET GENERICLU** *lu* **FAILED DUE TO ACTIVE VTAM SESSIONS**

Explanation: The RESET GENERICLU command was not accepted because one or more active VTAM sessions exist to the indicated LU. All VTAM sessions must be terminated before the command is accepted.

lu The value supplied on the RESET GENERICLU command

System action: The command is rejected.

System programmer response: Terminate the VTAM sessions to the remote LU and reissue the -RESET GENERICLU command.

DSNL040E *csect-name* **RESET GENERICLU** *lu* **FAILED DUE TO INDOUBT RESOLUTION INFORMATION**

Explanation: The RESET GENERICLU command was not accepted because information is currently being retained for resolution of indoubt threads for the indicated partner LU. The DISPLAY THREAD TYPE(INDOUBT) command can be used to view this information.

lu The value supplied on the RESET GENERICLU command

System action: The command is rejected.

System programmer response: Make sure that all indoubt threads at the remote site are resolved before issuing the RESET GENERICLU command.

When DB2 performs automatic indoubt thread resolution, the information contained in the DISPLAY THREAD TYPE(INDOUBT) display is purged at the end of the resynchronization process.

If required, use the -RESET INDOUBT command to manually purge the indoubt resolution information.

DSNL041E *csect-name* **DB2 GENERIC LU NAME REQUESTED, BUT REQUIRED VTAM LEVEL IS NOT INSTALLED**

Explanation: The DB2 GENERIC LUNAME installation option was specified. This option requires VTAM Version 4 Release 2 or a later release.

System action: The DB2 GENERIC LUNAME installation option is ignored.

System programmer response: If the DB2 GENERIC

LUNAME is required, install VTAM Version 4 Release 2 or a later release.

DSNL042E *csect-name* **DB2 SYSPLEX TP WAS INVOKED BY LU** *lu*, **BUT REQUIRED MVS LEVEL IS NOT INSTALLED**

Explanation: A request for the DB2 sysplex transaction program (TP) was received by a DB2 server that is a member of a data sharing group. The DB2 sysplex TP is used by the requesting systems to determine which DB2 systems within the group are currently available for DDF conversations.

The DB2 sysplex TP uses Work Load Manager (WLM) functions in MVS Version 5 Release 2 or a later release to balance distributed workload among the DB2 systems in the group. Because the required MVS release is not installed or DB2 is not registered with WLM, the DB2 sysplex TP is not able to provide the requested information.

lu The name of the LU in the network that invoked the DB2 sysplex TP

System action: The DB2 sysplex TP returns the NETID.LUNAME of the local DB2 system as the only available DB2 server in the group. The requester is not provided the names of other DB2 systems in the group, so all sessions from the requester are routed to the local DB2 server.

System programmer response: Install MVS Version 5 Release 2 or a later release if you want to have the DB2 sysplex TP return multiple NETID.LUNAME values to the requester or refer to the DSNL044I on why DB2 could not register with WLM.

DSNL043I *csect-name* **RESET GENERICLU** *luname* **INVALID**

Explanation: The luname value specified is invalid. The correct format is either a

- 1 to 8 character luname, or
- 1 to 8 character network ID, a period, and a 1 to 8 character luname.

System action: The parameter is ignored. Processing continues with the next luname, if any.

User response: Determine the correct luname and reissue the command.

DSNL044I *csect-name* **MVS** *macname* **MACRO FAILED WITH RETCODE=***xx* **AND RSNCODE=***yy*

Explanation: An MVS Workload Manager (WLM) macro returned an unexpected return code, which is described by the indicated RETCODE and RSNCODE values. DDF uses MVS Workload Manager macros to obtain data sharing sysplex routing information for incoming DDF work requests, and to establish an MVS enclave for the execution environment of the distributed thread.

macname
The name of the MVS macro that returned the unexpected return code. The possible values are:

- IWMSRSRG: Register with WLM for sysplex routing.
- IWMSRSRS: Request sysplex routing information.
- IWMSRDRS: Deregister with WLM for sysplex routing.
- IWMECREA: Create enclave.

The WLM sysplex routing service macros will be appended with the contents of the LOCATION parameter passed to it. The contents of the LOCATION parameter will either be the member's location name or one of the alias-names being defined to subset the data sharing group. If the member's location name is displayed, then this member will not participate in WLM workload balancing. If a subsetting alias-name is displayed, then subsetting of the data sharing group by that alias-name will be disabled. However, a connection to this member of the data sharing group will be accepted when using the alias-name instead of the member's location name.

xx The hexadecimal value of the MVS RETCODE field.

yy The hexadecimal value of the MVS RSNCODE field.

System action: If a sysplex routing macro failed, then sysplex routing may not be performed in which case, all SQL statements received from the client will be processed by this DB2 subsystem. If an enclave macro failed, the distributed thread may not be able to execute in an MVS enclave, however the distributed thread will continue to execute as an MVS SRB.

System programmer response: Refer to *MVS/ESA Programming: Workload Management Services* for the meaning of the MVS RETCODE and RSNCODE values.

DSNL045I *csect-name* **ICSF** *macname* **FUNCTION FAILED WITH RETCODE=***return-code* **AND RSNCODE=***reason-code*

Explanation: An Integrated Cryptographic Service Facility (ICSF) call returned an unexpected return code, which is described by the indicate RETCODE and RSNCODE values. DDF uses ICSF calls to create connection keys, to generate a shared private key, to encrypt data, and to decrypt data.

macname
The name of the ICSF call that returned the unexpected return code. The possible values are:

- CSNBRNG: Random Number Generate.
- CSNBCKM: Multiple Clear Key Import.
- CSNDPKB: Key Token Build.
- CSNDPKE: Key Encrypt.
- CSNBENC: Encipher data.
- CSNBDEC: Decipher data.

return-code
The hexadecimal value of the ICSF return_code field.

reason-code
The hexadecimal value of the ICSF reason_code field.

System action: The encryption function cannot be processed.

System programmer response: System Programmer Response: If the encryption facility is not installed, then install it before requesting encryption. If the encryption facility is installed, then verify that it is working correctly. Refer to *z/OS Integrated Cryptographic Service Facility - Application Programmer's Guide* - ICSF Return Codes and ICSF Reason Codes for further information.

DSNL046I *csect-name* **ICSF is not enabled**

Explanation: This message indicates that a cryptographic service is required, but the Integrated Cryptographic Service Facility (ICSF) is unavailable.

System action: If the following conditions are true, DB2 flows the user ID and password in the clear and processing continues:

- DB2 is acting as a requester
- The application is connecting to a remote site using TCP/IP and in the SYSIBM.IPNAMES table, the SECURITY_OUT value is 'P'.

Otherwise, the request fails.

System programmer response: If the cryptographic service facility is not installed, install it before requesting encryption functions. Refer to z/OS Cryptographic Services - Integrated Cryptographic Service Facility - System Programmer's Guide for more information.

DSNL065I *csect-name* **CURRENT DDF STATE DOES NOT PERMIT STOP DDF MODE(SUSPEND)**

Explanation: The current DDF state does not allow suspend processing to be initiated. DDF must be started for normal processing in order for suspend processing to be initiated.

System action: The command is not executed.

DSNL066I *csect-name* **STOP DDF MODE(SUSPEND) COMPLETE**

Explanation: DDF suspend processing has completed successfully. DDF has suspended all inbound activity.

System action: Processing continues normally.

DSNL067I *csect-name* **CURRENT DDF STATE DOES NOT PERMIT START DDF**

Explanation: The current DDF state does not allow START DDF processing to be initiated.

System action: The command is not executed.

DSNL068I *csect-name* **START DDF (RESUME PROCESSING) COMPLETE**

Explanation: DDF has resumed normal processing.

System action: Processing continues normally.

DSNL069I *csect-name* **DDF IS SUSPENDING**

Explanation: DDF has initiated suspend processing.

System action: Processing continues normally.

DSNL070I *csect-name* **DDF IS RESUMING**

Explanation: DDF has initiated resume processing.

System action: Processing continues normally.

DSNL071I *csect-name* **WAIT OR CANCEL REQUIRES KEYWORD SUSPEND**

Explanation: The WAIT(*n*) or CANCEL(*n*) keywords require that MODE(SUSPEND) be specified. For information about the STOP DDF command, see *DB2 Command Reference*.

System action: The command is not executed.

DSNL072I *csect-name* **WAIT TIME EXPIRED, DDF RESUME PROCESSING INITIATED**

Explanation: The DDF suspend process has not completed successfully in the period of time specified for the WAIT keyword, so DDF resume processing has been initiated. For information about the STOP DDF command, see *DB2 Command Reference*.

System action: DDF resume processing has been initiated.

DSNL073I *csect-name* **CANCEL TIME EXPIRED, CANCEL ACTIVE DBAT PROCESSING HAS BEEN INITIATED**

Explanation: The DDF suspend process has not completed successfully in the period of time specified for the CANCEL keyword so DDF is cancelling all active data base access threads (DBATS). For information about the STOP DDF command, see *DB2 Command Reference*.

System action: DDF is attempting to terminate all active database access threads.

DSNL074I **DSNLEDDA - DB2 DIST BELOW-THE-BAR STORAGE WARNING** *sysconsume* **% CONSUMED**

Explanation: The ssnmDIST address space is consuming a large amount of the available 31-bit storage below the bar. *sysconsume%* of the available 31-bit storage below the bar is consumed by DB2.

System programmer response: Consider reducing the workload on the system.

Programmer response: If possible, modify the user application identified in the subsequent DSNL077I message to commit more frequently and close cursors earlier. This allows DB2 to release resources and free up virtual storage. If you suspect an error in DB2, refer to Part 2 of DB2 Diagnosis Guide and Reference for information on identifying and reporting the problem.

DSNL075I **DSNLEDDA - BEGIN DISPLAY OF LARGEST STORAGE CONSUMERS IN DIST**

Explanation: A summary of the active DBATs that consume the largest amount of 31-bit storage below the bar in the ssnmDIST address space follows in a series of DSNL077I messages

DSNL076I **DSNLEDDA - END DISPLAY OF LARGEST STORAGE CONSUMERS IN DIST**

Explanation: A summary of the active DBATs that consume the largest amount of 31-bit storage below the bar in the ssnmDIST address space precedes this message.

DSNL077I **DSNLEDDA - AGENT** *rank***:**

```
# NAME: name ST: status A:act-ind
# REQ req-ct  corr-id AUTHID: auth-id
# PLAN: pname ASID: asid TOKEN: token
# STG: thread-stg LUW-ID: luw-id=token
# THREAD-INFO: thread-info LOCATION: location
```

Explanation: The agent described in this message is the greatest consumer of 31-bit storage below the bar in the ssnmDIST address space. See the description of message DSNV404I for explanations of fields *name*, *status*, *act-ind*, *req-ct*, *corr-id*, *auth-id*, *pname*, *asid*, and *token*. The field *thread-stg* is used for serviceability. The remaining fields *luw-id=token*, *thread-info*, and *location* provide more information for identifying the thread consuming excessive DIST storage.

luw-id=token
The logical-unit-of-work ID of the thread followed by an '=' sign and a 5 digit token value which can be used in any DB2 command that accepts luw-id as input.

thread-info
Thread-info contains additional identifiers that define more specific information regarding the identity of the thread. In many distributed configurations, the primary authorization-ID used with DB2 is not necessarily the user-ID of the end user. The following identifiers, separated by a colon(:), are provided:

- The primary authorization-ID associated with the thread.
- Work station name of the end user's work station. An asterisk (*) indicates that it is not relevant to the thread or was not provided by the client.
- End user's user-ID. An asterisk (*) indicates that it is not relevant to the thread or was not provided by the client.
- Application name of the end user's application. An asterisk (*) indicates that it is not relevant to the thread or was not provided by the client.

location Identifies the requesting site. The value specified in location is one of the following:

- The relational database name of the requester or the VTAM LU name of the requester *(luname)*, a colon (":") delimiter, and the LU name of the requester for SNA connections.
- The IP address of the requester for TCP/IP connections.

Operator response: To cancel the agent and gather a diagnostic dump, a -CANCEL THREAD (*token*) DUMP command can be issued.

DSNL080I *csect* **DISPLAY DDF REPORT FOLLOWS:**

Explanation: This message is issued as a result of the DISPLAY DDF command. This message is always generated as the title for the DISPLAY DDF command report output. It precedes the other messages generated by this command.

System action: Processing continues normally.

DSNL081I STATUS=*status*

Explanation: This message is issued as a result of the DISPLAY DDF command. This message displays one of the following DDF statuses:

STARTD
Started

STARTG
Starting

STOPDF
Stopped Force. Last stopped abnormally, such as by "Stop DDF Mode(Force)" command

STOPDQ
Stopped Quiesce. Last stopped normally, such as by the "Stop DDF Mode(Quiesce)" command or DDF has not yet been started

STOPGF
Stopping Force. Currently stopping abnormally, such as by the "Stop DDF Mode(Force)" command

STOPGQ
Currently stopping normally, such as by "Stop DDF Mode(Quiesce)" command

SUSPND
Suspended. Currently suspended, such as by the "Stop DDF Mode(Suspend)" command

SUSPNG
Suspending. Currently suspending, such as by the "Stop DDF Mode(Suspend)" command

NOINIT
Not initialized. DDF is not configured. A "DDF STARTUP OPTION" value of "NO" is specified in the DSNTIPR installation panel.

System action: Processing continues normally.

DSNL082I LOCATION LUNAME GENERICLU

Explanation: This message is issued as a result of the DISPLAY DDF command. This message displays the column headings for the DSNL083I message that follows

System action: Processing continues normally.

DSNL083I *location luname genericlu*

Explanation: This message is issued as a result of the DISPLAY DDF command. This message provides the column values for hte headings described by the DSNL082I message.

location
The location name of DDF as defined by the "DB2 LOCATION NAME" value in the DSNTIPR installation panel. Also, it is the location value of the BDSD DDF record that is defined by the 'LOCATION' value in the DDF satatement of the DSNJU003 Change Log Inventory utility. A value of 'NONE' indicates that a location name has not been configured.

This information also appears in mssage DSNL004I when DDF is started. See the DSNL004I message for additional information.

luname
The DDF LU name. You can determine the LU name from the 'DB2 NETWORK LUNAME' value in the DSNTIPR installation panel. You may also determine the LU NAME from the LU name value of the BSDS DDF record that is defined by the 'LUNAME' value in the DDF statement of the DSNJU003 Change Log Inventory utility. A value of '-NONE' indicates that an LU name has not been configured.

If an LU name has been configured, then the field will contain the fully qualified LU name that consists of a network ID, a period delimiter, and the LU name. The network ID is determined from VTAM once DDF is started. A network ID of '-NONE' indicates that DDF has not yet ben started, or initialization with VTAM has failed.

This information also appear in message DSNL004I when DDF is started. See the DSNL004I message for additional information.

genericlu
The DDF generic LU name. The generic LU name is determined from the 'DB2 GENERIC LUNAME' value in the DSNTIPR installation panel, and the generic LU name value of the BSDS DDF record which is defined by the 'GENERIC' value in the DDF statement of the DSNJU003 Change Log Inventory utility. A value of '-NONE' indicates that a generic LU name has not been configured.

If a generic LU name has been conifgured, then the field will contain the fully qualified generic LU name which consists of a network ID, a period delimiter, and the generic LU name. The network ID is determined from VTAM once DDF is started. A network ID of '-NONE' indicates thtat DDF has not yet been started, or initialization with VTAM has failed.

This information also appears in message DSNL004I when DDF is started. See the DSNL004I message for additional information.

System action: Processing continues normally.

DSNL084I IPADDR TCPPORT RESPORT

Explanation: This message is issued as a result of the DISPLAY DDF command. This message displays the column heading for the DSNL085I message that follows.

System action: Processing continues normally.

DSNL085I *ip-address tcpport resport*

Explanation: This message is issued as a result of the DISPLAY DDF command. This message provides the column values for the headings described by the DSNL084I message.

ip-address
The IP address of DDF.

During DDF start processing, DDF issues a GetHostID (z/OS UNIX System Services BPX1HST macro) socket call to determine its IP address if the listener ports are defined. A value of '-NONE' indicates that DDF is not configured to use TCP/IP network protocols (see tcpport), DDF has not yet been started, or TCP/IP initialization has failed.

tcpport The SQL listener TCP/IP port number.

The SQL listener port is determined from the 'DRDA PORT' value in the DSNTIP5 installation panel, and the port value of the BSDS DDF record. The 'PORT' value in the DDF statement of the DSNJU003 Change Log Inventory utility defines the port value of the BSDS DDF record.

- A value of 0 indicates that an SQL listener port has not been specified, and DDF is not configured to use TCP/IP network protocols.
- A value of 'NONE' indicates that DB2 will not accept connections on this port. This may be due to specifying 0 for the DSN6SYSP MAXDBAT installaiton parameter defined in DSNZPARM. For more information, see the 'MAX REMOTE ACTIVE' specification in the DSNTIPE installation panel in the DB2 Installation Guide. The current 'MAX REMOTE ACTIVE' value can also be seen in theMDBAT' value of the DSNL090I message that is included in the 'DISPLAY DDF DETAIL' report.

This information also appears in message DSNL004I when DDF is started. See the DSNL004I message for additional information.

resport The two phase commit resynchronization (resync) listener TCP/IP port number.

The resync listener port is determined from the 'RESYNC PORT ' value in the DSNTIP5 installation panel, and the resynchronization port value of the BSDS record. The 'RESPORT' value in the DDF statement of the DSNJU003 Change Log Inventory utility defines the resynchronization port value of the BSDS DDF record. A value of 0 indicates that a resyncronization listener port has not been specified, and DDF is not configured to use TCP/IP network protocols.

This information also appears in message DSNL004I when DDF is started. See the DSNL004I message for additional information.

System action: Processing continues normally.

DSNL086I *domain-type* **DOMAIN=***domain-name*

Explanation: This message is issued as a result of the DISPLAY DDF command. This message displays a TCP/IP domain name associated with DDF.

domain-type
The type of domain name that is displayed, which can be one of the following:

SQL The domain name is used to accept inbound SQL requests from remote partners. The *tcport* value of message DSNL085I message provides the port number that is used to accept inbound connections.

RESYNC
The domain name is used to accept inbound two-phase commit resynchronization requests from remote partners. The *resport* value of message DSNL085I provides the port number that is being used to accept inbound connections.

Normally, the SQL and RESYNC domain names will be the same. However, in data sharing environments, the SQL domain name will be different than the RESYNC domain name. See *DB2 Data Sharing: Planning and Administration* for more information.

domain-name
The domain name associated with the domain type. A value of "-NONE" indicates that the use of TCP/IP is not configured or that TCP/IP initialization has failed.

This information also appears in message DSNL519I and/or message DSNL004I when DDF is started. See the DSNL519I and DSNL004I messages for additional information.

System action: Processing continues normally.

DSNL090I **DT=***dt* **CONDBAT=***condbat* **MDBAT=***mdbat*

Explanation: This message is issued as a result of the DISPLAY DDF DETAIL command. This message displays some of the system parameter information associated with DDF.

dt DDF thread value. It indicates the "DDF THREADS" value specified in the DSNTIPR installation panel.

- A - DDF is configured with DDF THREADS ACTIVE
- I - DDF is configured with DDF THREADS INACTIVE

condbat
Maximum number of inbound connections as determined by the "MAX REMOTE CONNECTED" value in the DSNTIPE installation panel.

mdbat Maximum number of database access threads as determined by the "MAX REMOTE ACTIVE" value in the DSNTIPE installation panel. This effectively determines the maximum number of active *slots*. That is, the maximum number of concurrent active database access threads that could potentially be executing SQL.

System action: Processing continues normally.

DSNL092I ADBAT=*adbat* **QUEDBAT=***quedbat* **INADBAT=***inadbat* **CONQUED=***conqued*

Explanation: This message is issued as a result of the DISPLAY DDF DETAIL command.

adbat Current number of database access threads (DBATs). This includes DBATs that are actively associated to work for remote connections and "disconnected" DBATs (see DSNL093I *dscdbat* for more information). This value increases as new DBATs get created or when inactive DBATs (see *inadbat* description for more information) become active. The value decreases as DBATs terminate or become inactive (see *inadbat*), or as "disconnected" DBATs (see DSNL093I *dscdbat*) terminate. When this value reaches or exceeds the DSNL090I *mdbat* value, then newly attached connections, or new work on existing inactive connections (see DSNL093I *inaconn* for more information) or threads (*inadbat*), might need to be queued (see *quedbat* for more information) if there are no "disconnected" DBATs (see DSNL093I *dscdbat*) available that can be used to service the work.

quedbat
This value reflects a cumulative counter that is always incremented when the DSNL090I *mdbat* limit has been reached. The *quedbat* value is equal to the cumulative number of newly attached connections, or new work on inactive connections (see DSNL093I *inaconn*), or new work on inactive DBATs (see *inadbat*) that had to wait for a DBAT to become available to service the new work. This value is identical to the QDSTQDBT statistical value and a non-zero value suggests that performance and throughput might have been affected, in which case it might be appropriate to consider increasing the *maxdbat* value. See *adbat* and DSNL090I *mdbat* for additional information. Also note that the *quedbat* counter is only reset at restart for this DB2 subsystem.

inadbat
Current number of type 1 inactive threads. Only applies if the "dt" value specified in the DSNL090I message indicates that DDF INACTIVE support is enabled. Any database access threads reflected here can also be observed in the DISPLAY THREAD TYPE(INACTIVE) command report.

conqued
Current number of connection requests that have been queued and are waiting to be serviced. Only applies if the "dt" value specified in the DSNL090I message indicates that DDF INACTIVE support is enabled.

These connections can represent newly attached connections, or inactive connections for which a new request has arrived from the client (connections that were represented in the *inaconn* value of the DSNL093I message).

If the maximum number of database access threads has been reached (adbat value is equal to or greater than the DSNL090I mdbat value) and there are no disconnected database access threads to service the request (see DSNL093I dscdbat) value, or if DDF is suspended or suspending (see DSNL081I status value), then these requests cannot yet be serviced.

If DDF is suspending or stopped suspended (see DSNL081I status value), then these conqued connections will not be processed until DDF is started (resumed). If the adbat value is equal to or greater than the DSNL090I mdbat value, then these conqued connections will not be processed until another active database access thread releases its slot by either terminating or by moving to the inactive state (see the inadbat or DSNL093I *inaconn* value).

System action: Processing continues normally.

| **DSNL093I DSCDBAT=***dscdbat* **INACONN=***inaconn*

Explanation: This message is issued as a result of the DISPLAY DDF DETAIL command.

dscdbat
Current number of disconnected database access threads. This value only applies if the *dt* value specified in the DSNL090I message indicates that DDF INACTIVE support is enabled. This value is effectively the number

of DBAT pool threads. DBAT pool threads are the database access threads that are available to service queued connections requests (see DSNL092I *conqued* value). Database access threads reflected here are also accounted for in the DSNL092I *adbat* value.

Over time, disconnected DBATs can hold allocation duration locks, such as locks related to DDL. These disconnected DBATs can be terminated by the STOP DDF MODE SUSPEND command, which allows their held allocation duration locks to be released. Stopping DDF with mode SUSPEND allows these disconnected DBATs to be terminated without affecting current or new connections, although activity on these current or new connections are not processed until DDF is started (resumed). You can also automatically terminate disconnect DBATs when they have been unused for a certain length of time, thus releasing their allocate duration locks. You can set this automatic termination by adjusting the DBAT pool thread timeout value as described by "POOL THREAD TIMEOUT" in the DSNTIP5 installation panel.

inaconn
Current number of inactive (previously called type 2) connections. This value only applies if the *dt* value specified in the DSNL090I message indicates that DDF INACTIVE support is enabled. This value represents connections with clients where the client last ended a unit of work (commit or rollback), causing DB2 to change the state of the connection to inactive. That means that the DBAT is separated from the connection, and the connection becomes inactive while the DBAT is made available to service new client connections or to service new requests on existing client connections. This value is not included in the DSNL092I *inadbat* value. When a new request arrives from the client system on the connection, the connection is then moved to the DSNL092I *conqued* value. Any connections reflected here can also be observed in the DISPLAY THREAD TYPE(INACTIVE) command report.

System action: Processing continues normally.

DSNL099I *csect* **DISPLAY DDF REPORT COMPLETE**

Explanation: This message indicates that the DISPLAY DDF report successfully completed.

System action: Processing continues normally.

DSNL200I DISPLAY LOCATION REPORT FOLLOWS-

Explanation: This message is the title for the -DISPLAY LOCATION command report output. It precedes the other messages generated by this command. This message is followed by messages DSNL201I and DSNL202I. The final message for the report is be message DSNL205I.

System action: The -DISPLAY LOCATION command continues normally.

DSNL201I LOCATION PRDID LINKNAME REQUESTERS SERVERS CONVS

Explanation: The column headings for the list of locations generated by the -DISPLAY LOCATION command are displayed. The definitions are listed below:

Heading
Description

LOCATION
The location name associated with the remote site

PRDID Identifies the database product at the location

LINKNAME
The VTAM LU name associated with the remote location

REQUESTERS
The number of agents at this site that are requesting data from the remote location

SERVERS
The number of agents at this site that are acting as a server to the remote location

CONVS
The total number of conversations in use between this site and the remote location

See message DSNL202I a for more specific description of these items.

The message number and subsystem recognition character are not included in this message.

System action: Processing continues normally.

DSNL202I LOCATION PRDID LINKNAME REQUESTERS SERVERS CONVS

Explanation: This message lists the status of communications between a remote location and this DB2 subsystem. Information about the status appears in the following list:

Item **Description**

LOCATION
A 16-byte name for the remote site.

PRDID Identifies the database product at the location in the form nnnvvrrm.

- nnn - Identifies the database product (e.g., DSN=DB2).
- vv - Product version.
- rr - Product release.
- m - Product modification level.

LINKNAME
The VTAM LU name or dotted decimal IP address associated with the remote location.

REQUESTERS
A count of the threads at this location that access data at the remote location. The count includes all the allied threads associated with the remote location, and any database access threads that are accessing data at the remote location.

SERVERS
A count of the database access threads at this location that access data for agents at the remote location.

CONVS
A count of the total number of conversations between the two systems. The count includes both allied and database threads.

The message number and subsystem recognition character are not included in this message.

System action: Processing continues normally.

DSNL203I L203-SYSTASK SESSID A ST TIME

Explanation: This message forms the heading for the listing of DB2 system threads that communicate with other locations. It follows each occurrence of message DSNL202I if the DETAIL keyword of the DISPLAY LOCATION command is used. This message is followed by one or more occurrences of message DSNL204I, which describe the DB2 system threads communicating with the remote site and its conversation information. Items in this part of the heading are:

SYSTASK
Heading that describes the DB2 system thread

SESSID
Heading for the associated conversation identifiers

A Heading for the associated active indicators

ST Heading for the associated status indicators

TIME Heading for the associated timestamps

See message DSNL204I for descriptions of these items.

The message number and subsystem recognition character are not included in this message. However, L203 is included in the message text to identify the message.

System action: Processing continues normally.

DSNL204I L204-SYSTASK SESSID ACTIVE STATUS TIME

Explanation: This message follows the DSNL203I message for a remote location when a detail display is specified on the DISPLAY LOCATION command. This message is displayed for each conversation owned by a DB2 system thread that communicates with the site. It lists detailed information about the conversations associated with the DB2 system thread.

systask Describes the DB2 system threads as follows:

- SYSCON-x

  Where x =

  O The system thread that manages 'outbound' DB2 system conversations.

  I The system thread that manages 'inbound' DB2 system conversations.

  When DB2 is communicating with other DB2 locations using system-directed access, DB2 establishes 'system conversations' to the remote DB2 location. DB2 uses these conversations internally. The conversations created are one directional. That is, DB2 has a conversation that it uses to send information to the remote site, and another that it uses to receive information from the remote site. A DB2 system thread is created to manage 'outbound' (used for sending) conversations and another to manage 'inbound' (used for receiving) conversations.
- RESYNC

  A conversation used to resynchronize a unit of work that encountered a previous thread or communication error. There can be one entry representing each thread needing resynchronization with the remote site.

sessid Two possible values can appear in this column:

1. For VTAM network connections, this column contains the VTAM-defined session instance identifier of the session on which the conversation is executing. If the session identifier is not applicable, this field contains zeros.

   VTAM does not supply the entire session id to DB2. The first two digits of the session id are incorrect. Therefore, the session id, as presented in this message, is

not acceptable in VTAM commands. Use the VTAM DISPLAY NET,ID=db2-luname,SCOPE=ACT command to obtain the full session ID. The DISPLAY NET command lists all sessions (SID) for the DB2 logical unit. Scan the DISPLAY NET output for the complete SID.

2. For TCP/IP network connections, this column contains a string of the form 'local:remote', where:

   local -- specifies the local DB2 system's TCP/IP port number

   remote -- specifies the remote partner's TCP/IP port number

active One of the following:

V The conversation is active within VTAM or TCP/IP.

W The conversation is suspended in DB2 waiting for VTAM or TCP/IP notification that the function is complete.

Blank, otherwise.

status Status of the conversation:

A Conversation in allocation.

C Session limits are being negotiated with the partner (CNOS) prior to conversation allocation.

D Conversation in deallocation.

R Receiving.

S Sending.

Blank otherwise.

time The timestamp (yydddhhmmssth) of the last message sent or received on the conversation.

The message number and subsystem recognition character are not included in this message. However, L204 is included in the message text to identify the message.

System action: Command processing continues

DSNL205I DISPLAY LOCATION REPORT COMPLETE

Explanation: This message follows messages DSNL200I, DSNL201I, and DSNL202I and indicates that the -DISPLAY LOCATION report successfully completed.

The message number and subsystem recognition character are not included in this message.

System action: Processing continues normally.

DSNL206I *csect-name named-command* **COMMAND INVALID, DDF NOT STARTED**

Explanation: The 'named-command' command was issued before the distributed data facility (DDF) was started. The 'named-command' command executes successfully only if the DDF is operating.

System action: Processing continues normally.

DSNL207I DISPLAY LOCATION NO LOCATIONS FOUND

Explanation: The -DISPLAY LOCATION command was unable to find any locations associated with the specified list of locations. If all locations were specified, there is no distributed activity.

System action: Processing completed normally.

User response: If a location list was specified, reissue the -DISPLAY LOCATION command using the LOCATION(*) parameter to display information for all locations, if desired.

DSNL208I DISPLAY LOCATION TERMINATED WITH MAX LINES

Explanation: The message indicates the -DISPLAY LOCATION report was truncated. The report was generated in response to a command from an MVS console and more than 254 response messages were generated. Only 254 response messages are returned.

System action: Command processing completes. The truncated report is written to the MVS console.

User response: Reissue the -DISPLAY LOCATION request specifying a specific location or list of locations.

DSNL400E INDOUBT THREAD RESOLUTION WITH COORDINATOR *location* **SHOWS HEURISTIC DAMAGE FOR** *luwid=token* **- REMOTE COMMIT VERSUS LOCAL ABORT. INFORMATION RECORDED IN TRACE RECORD WITH IFCID=207 AND IFCID SEQUENCE NUMBER=***seq-no1* **AND IN TRACE RECORD WITH IFCID=203 AND IFCID SEQUENCE NUMBER=***seq-no2*

Explanation: DB2 has participant responsibility in the logical unit of work identified by 'luwid=token'. Resolution with the coordinator identified by 'location' shows heuristic damage occurred. The DB2 thread representing the logical unit of work was previously resolved using the RECOVER INDOUBT command. The action taken was to abort the thread. Subsequently, communication was reestablished with the coordinator and resolution shows that the coordinator committed the thread.

- The application was told that commit succeeded.

- Any resources updated at the coordinator were committed.
- Updated resources at this DB2 were rolled back.

location
The field contains one of the following:
- The relational database name (location name)
- The VTAM LU name (<luname>)
- The dotted decimal IP address (nnn.nnn.nnn.nnn).

luwid=token
The global logical unit of work id and the local token that identify the thread.

seq-no1
The IFCID sequence number associated with the 207 trace record. The sequence number is printed in hexadecimal format. If the sequence number is zero, IFCID 207 was not active or it was active with a destination of the resident trace table only. IFCID 207 is a member of statistics class 4, which has a default destination of SMF. This trace record is written when the heuristic damage condition was detected and contains information about the damage condition.

seq-no2
The IFCID sequence number associated with the 203 trace record. The sequence number is printed in hexadecimal format. If the sequence number is zero, IFCID 203 was not active or it was active with a destination of the resident trace table only. IFCID 203 is a member of statistics class 4, which has a default destination of SMF. This trace record was written when the heuristic decision to commit or abort was made and contains information about the decision, including the names of any downstream participants.

System action: Alert A004 is generated. A trace record whose IFCID=207 is written. Indoubt resolution with the coordinator completes.

System programmer response: Determine the destination of statistics class 4 and locate and print the trace record with SEQUENCE NUMBER='seq-no1'. Call the database administrator.

Database Administrator Action: Correct the damage. This is not an easy task. Since the time of the heuristic action, the data that was aborted with the RECOVER INDOUBT command might have been used by many applications - read and/or write. Correcting the damage might involve reversing the effects of these applications as well. Tools available are:
- The statistics trace class 4 contains an IFCID 207 trace record. This record contains the RBA of the first DB2 log record written for the logical unit of work whose LUWID is included in this message.
- The DSN1LOGP utility provides a summary report that identifies the table spaces modified by the logical unit of work and any downstream participant locations involved in the logical unit of work.
- DISPLAY THREAD TYPE(INDOUBT) displays the RBA of the first DB2 log record written for the logical unit of work whose LUWID is included in this message. This information continues to appear in the DISPLAY THREAD report until it is purged with the RESET INDOUBT command.

Contact your IBM Support Center for further assistance.

DSNL401E INDOUBT THREAD RESOLUTION WITH COORDINATOR *location* **SHOWS HEURISTIC DAMAGE FOR** *luwid=token* **- REMOTE ABORT VERSUS LOCAL COMMIT. INFORMATION RECORDED IN TRACE RECORD WITH IFCID=207 AND IFCID SEQUENCE NUMBER=***seq-no1* **AND IN TRACE RECORD WITH IFCID=203 AND IFCID SEQUENCE NUMBER=***seq-no2*

Explanation: DB2 has participant responsibility in the logical unit of work identified by 'luwid=token'. Resolution with the coordinator identified by 'location' shows heuristic damage occurred. The DB2 thread representing the logical unit of work was previously resolved with the RECOVER INDOUBT command. The action taken was to commit the thread. Subsequently, communication was reestablished with the coordinator and resolution shows that the coordinator aborted the thread.
- The application was told that commit failed.
- Any resources updated at the coordinator were rolled back.
- Updated resources at this DB2 were committed.

location
The field contains one of the following:
- The relational database name (location name)
- The VTAM LU name (<luname>)
- The dotted decimal IP address (nnn.nnn.nnn.nnn).

luwid=token
The global logical unit of work id and the local token that identify the thread.

seq-no1
The IFCID sequence number associated with the 207 trace record. The sequence number is printed in hexadecimal format. If the sequence number is zero, IFCID 207 was not active or it was active with a destination of the resident trace table only. IFCID 207 is a member of statistics class 4, which has a default

destination of SMF. This trace record is written when the heuristic damage condition was detected and contains information about the damage condition.

seq-no2
The IFCID sequence number associated with the 203 trace record. The sequence number is printed in hexadecimal format. If the sequence number is zero, IFCID 203 was not active or it was active with a destination of the resident trace table only. IFCID 203 is a member of statistics class 4, which has a default destination of SMF. This trace record was written when the heuristic decision to commit or abort was made and contains information about the decision, including the names of any downstream participants.

System action: Alert A004 is generated. A trace record whose IFCID=207 is written. Indoubt resolution with the coordinator completes.

System programmer response: Determine the destination of statistics class 4. Locate and print the trace record with SEQUENCE NUMBER='seq-no1'. Call the database administrator.

Database Administrator Action: Correct the damage. This is not an easy task. Since the time of the heuristic action, the data that was committed with the RECOVER INDOUBT command might have been used by many applications - read and/or write. Correcting the damage might involve reversing the effects of these applications as well. Tools available are:

- The statistics trace class 4 contains an IFCID 207 trace record. This record contains the RBA of the first DB2 log record written for the logical unit of work whose LUWID is included in this message.
- The DSN1LOGP utility provides a summary report that identifies the table spaces modified by the logical unit of work, and any downstream participant locations involved in the logical unit of work.
- DISPLAY THREAD TYPE(INDOUBT) displays the RBA of the first DB2 log record written for the logical unit of work whose LUWID is included in this message. This information continues to appear in the DISPLAY THREAD report until it is purged with the RESET INDOUBT command.

Contact your IBM Support Center for further assistance.

DSNL402I INDOUBT THREAD RESOLUTION WITH PARTICIPANT *location* SHOWS HEURISTIC DAMAGE FOR *luwid=token* - REMOTE COMMIT VERSUS LOCAL ABORT. INFORMATION RECORDED IN TRACE RECORD WITH IFCID=207 AND IFCID SEQUENCE NUMBER=*seq-no*

Explanation: DB2 has coordinator responsibility for the logical unit of work identified by 'luwid=token'. Resolution with a participant shows heuristic damage occurred. The participant identified by 'location' manually resolved the indoubt thread identified by 'luwid=token'. The action taken was to commit the thread. This is inconsistent with the abort decision of the coordinator.

- The application was told that commit failed.
- Any resources updated at the participant were committed.
- Updated resources at this DB2 were rolled back.

location
The field contains one of the following:

- The relational database name (location name)
- The VTAM LU name (<luname>)
- The dotted decimal IP address (nnn.nnn.nnn.nnn).

luwid=token
The global logical unit of work id and the local token that identify the thread.

seq-no The IFCID sequence number associated with the trace record. The sequence number is printed in hexadecimal format. If the sequence number is zero, then IFCID 207 was not active or it was active with a destination of the resident trace table only. IFCID 207 is a member of statistics class 4, which has a default destination of SMF.

System action: Alert A004 is generated. A trace record whose IFCID=207 is written. Indoubt resolution with the participant completes.

System programmer response: Call the database administrator.

Database Administrator Action: Inform the database administrator at 'location' that heuristic damage occurred for the logical unit of work with LUWID='luwid'. DB2 was the coordinator for the logical unit of work and made the decision to abort the database updates made by 'luwid'. At 'location' a heuristic decision was made to COMMIT the updates made by 'luwid'.

Information about the heuristic damage continues to appear in the DISPLAY THREAD TYPE(INDOUBT) report until it is purged with the RESET INDOUBT command.

DSNL403I INDOUBT THREAD RESOLUTION WITH PARTICIPANT *location* SHOWS HEURISTIC DAMAGE FOR *luwid=token* - REMOTE ABORT VERSUS LOCAL COMMIT. INFORMATION RECORDED IN TRACE RECORD WITH IFCID=207 AND IFCID SEQUENCE NUMBER=*seq-no*

Explanation: DB2 has coordinator responsibility for the logical unit of work identified by 'luwid=token'. Resolution with a participant shows heuristic damage occurred. The participant identified by 'location' manually resolved the indoubt thread identified by 'luwid=token'. The action taken was to abort the thread. This is inconsistent with the commit decision of the coordinator.

- The application was told that commit succeeded.
- Any resources updated at the participant were rolled back.
- Updated resources at this DB2 were committed.

location
The relational database name (location-name), SNA LU name (<luname>), or dotted decimal IP address of the participant.

luwid=token
The global logical unit of work id and the local token that identify the thread.

seq-no The IFCID sequence number associated with the trace record. The sequence number is printed in hexadecimal format. If the sequence number is zero, IFCID 207 was not active or it was active with a destination of the resident trace table only. IFCID 207 is a member of statistics class 4, which has a default destination of SMF.

System action: Alert A004 is generated. A trace record with IFCID 207 is written. Indoubt resolution with the participant completes.

System programmer response: Call the database administrator.

Database Administrator Action: Inform the database administrator at 'location' that heuristic damage occurred for the logical unit of work with LUWID='luwid'. DB2 was the coordinator for the logical unit of work and made the decision to COMMIT the database updates made by 'luwid'. At 'location' a heuristic decision was made to back out (ABORT) the updates made by 'luwid'.

Information about the heuristic damage continues to appear in the DISPLAY THREAD TYPE(INDOUBT) report until it is purged with the RESET INDOUBT command.

DSNL404E PROTOCOL ERROR DURING INDOUBT THREAD RESOLUTION WITH COORDINATOR *location* LUWID=*luwid=token* INFORMATION RECORDED IN TRACE RECORD WITH IFCID=206 AND IFCID SEQUENCE NUMBER=*seq-no*

Explanation: DB2 has participant responsibility for the logical unit of work identified by 'luwid=token'. The DB2 thread associated with the logical unit of work is indoubt. A protocol error occurred during indoubt resolution with the coordinator identified by 'location'.

location
Either the relational database name (location-name) or the logical unit name (<luname>) of the coordinator.

luwid=token
The global logical unit of work id and the local token that identify the thread.

seq-no The IFCID sequence number associated with the trace record. The sequence number is printed in hexadecimal format. If the sequence number is zero, IFCID 206 was not active or it was active with a destination of the resident trace table only. IFCID 206 is a member of statistics class 4, which has a default destination of SMF.

System action: The indoubt thread remains indoubt. Alert A003 is generated. A trace record whose IFCID=206 is written.

System programmer response: This is an APARable error, probably at the coordinator, but possibly at the DB2 participant. The DB2 documentation required is contained in the statistics class 4 trace record that was written. Determine the destination of statistics class 4. Locate and print the trace record with SEQUENCE NUMBER='seq-no'.

Call the database administrator.

Database Administrator Action: DB2 does not attempt to automatically resolve the indoubt thread. The indoubt thread must be manually resolved. The commit or abort decision made at the coordinator must be determined so that the same decision can be made at this participant DB2.

Contact the database administrator at the coordinator location to determine whether the thread with LUWID=luwid committed or aborted.

If the coordinator system is another DB2, the following steps can be taken at the DB2 coordinator to determine the commit or abort decision.

1. If the coordinator DB2 system is running and a DB2 conditional restart was not performed since the error was detected at this (the participant) location, ask the database administrator at the DB2

coordinator location to issue one of the following DISPLAY THREAD TYPE(INDOUBT) commands:

- A -DISPLAY THREAD TYPE(INDOUBT), using the LOCATION keyword and specifying the location name of the local DB2.
- A -DISPLAY THREAD TYPE(INDOUBT) LUWID(luwid)

If a thread with LUWID=luwid appears in the display, then the display also includes the commit or abort decision. If no thread with LUWID=luwid appears in the display, the thread was aborted.

2. If the DB2 coordinator is down, or if a conditional restart was performed there, ask the database administrator at the DB2 coordinator to run DSN1LOGP, specifying SUMMARY(ONLY) FILTER LUWID(luwid). The output from DSN1LOGP includes the commit or abort decision.

Use the RECOVER INDOUBT command at this (the participant) location to resolve the indoubt thread.

DSNL405I THREAD *luwid=token* **PLACED IN INDOUBT STATE BECAUSE OF COMMUNICATION FAILURE WITH COORDINATOR** *location***. INFORMATION RECORDED IN TRACE RECORD WITH IFCID=209 AND IFCID SEQUENCE NUMBER=***seq-no*

Explanation: During execution of the two phase commit protocol with the coordinator identified by 'location', communication failed. Phase 1 of the protocol completed and the thread is prepared for either commit or abort.

The thread is placed in the indoubt state and appears in a DISPLAY THREAD TYPE (INDOUBT) report. Periodic attempts are made to reestablish communication with the coordinator for automatic indoubt resolution. If successful, message DSNL407 or DSNL415 is written.

luwid=token
The global logical unit of work id and the local token that identify the thread.

location
The field contains one of the following:

- The relational database name (location name)
- The VTAM LU name (<luname>)
- The dotted decimal IP address (nnn.nnn.nnn.nnn).

seq-no The IFCID sequence number associated with the trace record. The sequence number is printed in hexadecimal format. If the sequence number is zero, IFCID 209 was not active or it was active with a destination of the resident trace table only. IFCID 209 is a member of statistics class 4, which has a default destination of SMF.

System action: Alert A006 is generated. A trace record whose IFCID=209 is written.

Operator response: If message DSNL407 or DSNL415 for the LUWID specified in the DSNL405 message does not appear in a reasonable amount of time, call the system programmer.

System programmer response: Determine the cause of the communication failure and have the problem fixed. DB2 periodically attempts to reestablish communication. If the problem cannot be fixed in a reasonable amount of time, call the database administrator. Manual resolution of the indoubt thread might be necessary to release locked resources.

Database Administrator Action: Take the following actions to resolve this thread using manual means (that is, take heuristic action) if it is necessary to release locks held by this thread.

- Determine the proper commit or abort decision by contacting the database administrator at the coordinator 'location' shown in this message in order to determine whether the thread with LUWID=luwid committed or aborted.

  If the coordinator system is another DB2, the following steps can be taken at the DB2 coordinator to determine the commit or abort decision.

  1. If the coordinator DB2 system is running and a DB2 conditional restart was not performed since the error was detected at this (the participant) location, ask the database administrator at the DB2 coordinator location to issue one of the following DISPLAY THREAD TYPE(INDOUBT) commands:
     - A -DISPLAY THREAD TYPE(INDOUBT), using the LOCATION keyword and specifying the location name of the local DB2.
     - A -DISPLAY THREAD TYPE(INDOUBT) LUWID(luwid)

     If a thread with LUWID=luwid appears in the display, then the display also includes the commit or abort decision. If no thread with LUWID=luwid appears in the display, then the thread was aborted.
  2. If the DB2 coordinator is down or if a conditional restart was performed there, ask the database administrator at the DB2 coordinator to run DSN1LOGP, specifying SUMMARY(ONLY) FILTER LUWID(luwid). The output from DSN1LOGP includes the commit or abort decision.
- Use the RECOVER INDOUBT command to commit or abort the thread.

DSNL406I THREAD *luwid=token* **AT PARTICIPANT** *location* **MAY BE INDOUBT BECAUSE OF A COMMUNICATION FAILURE. DB2 COMMITTED THE THREAD. INFORMATION RECORDED IN TRACE RECORD WITH IFCID=209 AND IFCID SEQUENCE NUMBER=***seq-no*

Explanation: During execution of the two phase commit protocol with the participant identified by 'location', communications failed. Phase 1 of the protocol completed with the participant, which is prepared to either commit or abort. The decision to commit the thread was made, but cannot be communicated to the participant at this time. The participant is indoubt.

DB2 becomes responsible for indoubt resolution with the participant. This responsibility appears in a DISPLAY THREAD TYPE (INDOUBT) report. Periodic attempts are made to reestablish communication with the participant for automatic indoubt resolution. If successful, message DSNL423 is written.

luwid=token
The global logical unit of work id and the local token that identify the thread.

location
The field contains one of the following:
- The relational database name (location name)
- The VTAM LU name (<luname>)
- The dotted decimal IP address (nnn.nnn.nnn.nnn).

seq-no The IFCID sequence number associated with the trace record. The sequence number is printed in hexadecimal format. If the sequence number is zero, IFCID 209 was not active or it was active with a destination of the resident trace table only. IFCID 209 is a member of statistics class 4, which has a default destination of SMF.

System action: Alert A006 is generated. A trace record whose IFCID=209 is written.

Operator response: If message DSNL423 for the LUWID specified in message DSNL406 does not appear in a reasonable amount of time, call the system programmer.

System programmer response: Determine the cause of the communication failure and have the problem fixed. DB2 periodically attempts to reestablish communication. If the problem cannot be fixed in a reasonable amount of time, call the database administrator. The participant might need to manually resolve the indoubt state to release locked resources.

Database Administrator Action: If the unit of work with LUWID=luwid is indoubt at 'location', inform the database administrator there that the logical unit of work with LUWID=luwid was committed to assist in manual resolution of the indoubt thread at 'location'.

DSNL407I AUTOMATIC RESOLUTION OF *luwid=token* **WITH COORDINATOR** *location* **RESULTED IN COMMIT**

Explanation: The identified indoubt logical unit of work was automatically resolved by communication with the coordinator location. After automatic resolution, this message is issued if message DSNL405 was previously issued to report the thread as indoubt. This message is not issued if message DSNL405 occurs in one restart of DDF or DB2, and automatic resolution occurs in a different restart of DDF or DB2.

luwid=token
The global logical unit of work id and the local token that identify the thread

location
The field contains one of the following:
- The relational database name (location name)
- The VTAM LU name (<luname>)
- The dotted decimal IP address (nnn.nnn.nnn.nnn).

System action: Processing continues normally.

DSNL408I INDOUBT THREAD RESOLUTION WITH COORDINATOR *location* **SHOWS HEURISTIC DAMAGE AT PARTICIPANT** *location* **FOR** *luwid=token* **- COORDINATOR COMMIT VERSUS PARTICIPANT ABORT. INFORMATION RECORDED IN TRACE RECORD WITH IFCID=207 AND IFCID SEQUENCE NUMBER=***seq-no*

Explanation: DB2 has participant responsibility in the logical unit of work identified by 'luwid=token' for the coordinator at COORDINATOR 'location' and coordination responsibility in the same logical unit of work for the participant at PARTICIPANT 'location'. Heuristic damage occurred.
- The application was told that commit succeeded.
- Any resources updated at the coordinator and at this DB2 were committed.
- Updated resources at the participant were rolled back.

location
The field contains one of the following:
- The relational database name (location name)
- The VTAM LU name (<luname>)
- The dotted decimal IP address (nnn.nnn.nnn.nnn).

luwid=token
The global logical unit of work id and the local token that identify the thread.

seq-no The IFCID sequence number associated with the trace record. The sequence number is printed in hexadecimal format. If the sequence number is zero, then IFCID 207 was not active or it was active with a destination of the resident trace table only. IFCID 207 is a member of statistics class 4, which has a default destination of SMF.

System action: Alert A004 is generated. A trace record whose IFCID=207 is written. Indoubt resolution with the coordinator completes.

System programmer response: Determine the destination of statistics class 4. Locate and print the trace record with SEQUENCE NUMBER='seq-no'. Call the database administrator.

Database Administrator Action: Inform the database administrator at the coordinator 'location' and at the participant 'location' of the problem.

Correcting the damage at the participant is not an easy task. Since the time of the heuristic action, the data that was aborted using manual means might have been used by many applications - read and/or write. Correcting the damage might involve reversing the effects of these applications as well. The following information that is available at DB2 might be helpful in correcting the problem:

- The statistics trace class 4 contains an IFCID 207 trace record. This record contains the RBA of the first DB2 log record written for the logical unit of work whose LUWID is included in this message.
- The DSN1LOGP utility provides a summary report that identifies the table spaces modified by the logical unit of work at the local DB2 as well as any downstream participant locations involved in the logical unit of work. This information, while not directly related to the damage at the participant, might be useful in determining the scope of the problem.
- DISPLAY THREAD TYPE(INDOUBT) displays the RBA of the first DB2 log record written for the logical unit of work whose LUWID is included in this message. This information continues to appear in the DISPLAY THREAD report until it is purged with the RESET INDOUBT command.

Contact your IBM Support Center for further assistance.

DSNL409I INDOUBT THREAD RESOLUTION WITH COORDINATOR *location* **SHOWS HEURISTIC DAMAGE AT PARTICIPANT** *location* **FOR** *luwid=token* **- COORDINATOR ABORT VERSUS PARTICIPANT COMMIT. INFORMATION RECORDED IN TRACE RECORD WITH IFCID 207 AND IFCID SEQUENCE NUMBER=***seq-no*

Explanation: DB2 has participant responsibility in the logical unit of work identified by 'luwid=token' for the coordinator at COORDINATOR 'location' and coordination responsibility in the same logical unit of work for the participant at PARTICIPANT 'location'. Heuristic damage occurred.

- The application was told that commit failed.
- Any resources updated at the coordinator and at this DB2 were rolled back.
- Updated resources at the participant were committed.

location
The field contains one of the following:

- The relational database name (location name)
- The VTAM LU name (<luname>)
- The dotted decimal IP address (nnn.nnn.nnn.nnn).

luwid=token
The global logical unit of work id and the local token that identify the thread.

seq-no The IFCID sequence number associated with the trace record. The sequence number is printed in hexadecimal format. If the sequence number is zero, then IFCID 207 was not active or it was active with a destination of the resident trace table only. IFCID 207 is a member of statistics class 4, which has a default destination of SMF.

System action: Alert A004 is generated. A trace record whose IFCID=207 is written. Indoubt resolution with the coordinator completes.

System programmer response: Determine the destination of statistics class 4. Locate and print the trace record with SEQUENCE NUMBER='seq-no'. Call the database administrator.

Database Administrator Action: Inform the database administrator at the coordinator 'location' and at the participant 'location' of the problem.

Correcting the damage at the participant is not an easy task. Since the time of the heuristic action, the data that was committed using manual means might have been used by many applications - read and/or write. Correcting the damage might involve reversing the effects of these applications as well. This information is

available at DB2 and might be helpful in correcting the problem.

- The statistics trace class 4 contains an IFCID 207 trace record. This record contains the RBA of the first DB2 log record written for the logical unit of work whose LUWID is included in this message.
- The DSN1LOGP utility provides a summary report that identifies the table spaces modified by the logical unit of work at the local DB2 and any downstream participant locations involved in the logical unit of work. This information, while not directly related to the damage at the participant, might be useful in determining the scope of the problem.
- DISPLAY THREAD TYPE(INDOUBT) displays the RBA of the first DB2 log record written for the logical unit of work whose LUWID is included in this message. This information continues to appear in the DISPLAY THREAD report until it is purged with the RESET INDOUBT command.

Contact your IBM support center for further assistance.

DSNL410I **WARM START CONNECTION BY PARTNER** *location* **REJECTED. PARTNER CHANGED AT LEAST 1 OF - OUR LOG NAME** *oldourname(newourname)*, **THEIR LOG NAME** *oldtheirname(newtheirname)*, **PA_SUPPORT** *oldpa(newpa)*, **FLAG_BYTE_SENT** *oldfb(newfb)*, **CCLUNAME_SENT** *oldccls(newccls)*, **AND INDOUBT THREADS REQUIRE RESOLUTION. SEE IFCID=205 TRACE RECORD WITH IFCID SEQUENCE NUMBER=***seq-no***.**

Explanation: An attempt to make a warm start connection with a partner was rejected because the partner specified a different set of sync point parameters than the ones that were in use when communications were lost. DB2 has knowledge of indoubt threads that involve the partner as either the coordinator or a participant. This error might be a recoverable error if the partner can restart with the original sync point parameters. If this is not possible, then the partner must perform a cold start connection with DB2.

location
The 'location' with which the connection failed: either the relational database name (location-name) or the logical unit name (<luname>) of the partner.

oldourname
The log name for this DB2 subsystem. This value is null if the partner did not initiate the log name exchange.

newourname
The log name for this DB2 subsystem, as remembered by our partner and sent to us in the attempted warm start connection. This value is null if the partner did not initiate the log name exchange.

oldtheirname
The log name used by the partner during the last connection, as remembered by DB2.

newtheirname
The log name used by the partner in the attempted warm start connection.

oldpa The sync point protocol used during the last connection, as remembered by the DB2 (PA - Presumed Abort, PN - Presumed Nothing).

newpa The sync point protocol used during the last connection, as remembered by the partner (PA - Presumed Abort, PN - Presumed Nothing).

oldfb The use of the PS header byte 2 during the last connection, as remembered by DB2 (F - used as flag byte, N - not used as flag byte).

newfb The use of the PS header byte 2 during the last connection, as remembered by the partner (F - used as flag byte, N - not used as flag byte).

oldccls Whether the luname of the conversation correlator was exchanged as part of the sync point protocol during the last connection, as remembered by DB2. (E - was exchanged, N - was not exchanged).

newccls
Whether the luname of the conversation correlator was exchanged as part of the sync point protocol during the last connection, as remembered by the partner (E - was exchanged, N - was not exchanged).

seq-no The IFCID sequence number associated with the trace record. The sequence number is printed in hexadecimal format. If the sequence number is zero, then IFCID 0205 was not active or it was active with a destination of the resident trace table only. IFCID 0205 is a member of statistics class 4, which has a default destination of SMF.

System action: The connection with the partner is rejected. Alert A002 is generated. DB2 retains indoubt knowledge.

System programmer response: Call the database administrator.

Database Administrator Action: Contact the database administrator at the partner 'location' and determine if it is possible for the partner to perform a warm start with same sync point parameters as ours ('oldourname', 'oldtheirname', 'oldpa', 'oldfb', 'oldccls'). If this is possible, the next attempt to connect will succeed.

If this cannot be done, then there are two other possible solutions:

- Have the partner 'location' perform a cold start connection with this DB2 subsystem.
- Use the RESET INDOUBT command with the FORCE option in conjunction with the LOCATION parameter to cause the next connection with the partner to be a cold start connection.

DSNL411E COLD START CONNECTION BY COORDINATOR *location* ACCEPTED. INDOUBT THREADS NEED MANUAL RESOLUTION. INFORMATION RECORDED IN TRACE RECORD WITH IFCID=204 AND IFCID SEQUENCE NUMBER=*seq-no*

Explanation: DB2 has participant responsibility for indoubt threads. The coordinator informed DB2 that it performed a cold start operation and lost all knowledge of indoubt threads. The indoubt threads at this DB2 must be manually resolved with the RECOVER INDOUBT command.

location
The field contains one of the following:

- The relational database name (location name)
- The VTAM LU name (<luname>)
- The dotted decimal IP address (nnn.nnn.nnn.nnn).

seq-no The IFCID sequence number associated with the trace record. The sequence number is printed in hexadecimal format. If the sequence number is zero, then IFCID 204 was not active or it was active with a destination of the resident trace table only. IFCID 204 is a member of statistics class 4, which has a default destination of SMF.

System action: The connection with the partner is accepted. Alert A001 is generated. A trace record whose IFCID=204 is written.

System programmer response: Call the database administrator.

Database Administrator Action: DB2 is a participant with one or more indoubt threads where the coordinator is 'location'. The DBMS at 'location' performed a cold start. The DB2 participant assumes that the coordinator recovery log was lost or damaged and indoubt thread resolution cannot be achieved. There is probably inconsistent data at the coordinator. A list of indoubt threads involving the coordinator 'location' can be displayed using the DISPLAY THREAD command for indoubt type threads by including the LOCATION('location') parameter in the command.

The heuristic decision (that is, to commit or abort the logical unit of work) should be coordinated with any other participants. The existence of other participants might not be easy to determine. The information might be available in the coordinators recovery log even though the coordinator performed a cold start. If the coordinator is a DB2, then it might be possible to execute the DSN1LOGP utility to produce a summary report. If this is possible, the following information might be produced:

- The LUWID of the thread.
- The NID used by IMS or CICS to identify the thread.
- The name of all participants.
- The name of an upstream coordinator if the coordinator is also a participant. This could happen when an application requester accesses a DB2 AS using application-directed access, which in turn accesses another DB2 server with system-directed access. (The local DB2 might be the third system in this scenario.)

At this DB2 participant, all indoubt threads coordinated by the cold starting location must be resolved with the RECOVER INDOUBT command. Locked data remains unavailable until resolution. Threads that were already resolved with this command are discarded. Threads not yet resolved are discarded after resolution with the command.

The commit or abort decision provided using the RECOVER INDOUBT command for a logical unit of work are propagated to all downstream participants, if any.

DSNL412I PROTOCOL ERROR DETECTED IN SYNCPOINT REQUEST/RESPONSE FROM COORDINATOR *location* LUWID=*luwid=token*. INFORMATION RECORDED IN TRACE RECORD WITH IFCID=208 AND IFCID SEQUENCE NUMBER=*seq-no*

Explanation: DB2 is a participant in the logical unit of work identified by 'luwid=token'. A protocol error occurred during the SNA sync point exchange with the coordinator identified by 'location'. The protocol error fits into one of the following categories:

- An unknown SNA PS header was received.
- An SNA PS header was received in the wrong sequence.
- An invalid LU6.2 verb sequence was used to deliver the SNA PS header.
- A DEALLOCATE TYPE(ABEND_SVC) was received from the coordinator, indicating that the coordinator detected a protocol violation in the PS headers transmitted by DB2.

location
Either the relational database name

(location-name) or the logical unit name (<luname>) of the coordinator.

luwid=token
The global logical unit of work id and the local token that identify the thread.

seq-no The IFCID sequence number associated with the trace record. The sequence number is printed in hexadecimal format. If the sequence number is zero, IFCID 208 was not active or it was active with a destination of the resident trace table only. IFCID 208 is a member of statistics class 4, which has a default destination of SMF.

If the protocol error was detected before the commit decision, the unit of work at the DB2 system might be indoubt. DB2 does not automatically resolve such an indoubt thread.

If the protocol error was detected after the commit decision, the unit of work either completed commit or abort processing.

System action: Alert A005 is generated. A trace record whose IFCID=208 is written.

System programmer response: This is an APARable error, probably at the coordinator, but possibly at DB2. The DB2 documentation required is contained in the statistics class 4 trace record. Determine the destination of statistics class 4. Locate and print the trace record with SEQUENCE NUMBER='seq-no'.

The database administrator might need to manually resolve the indoubt thread.

Database Administrator Action: Determine if the 'luwid' is indoubt by using the DISPLAY THREAD command for type indoubt threads and including the LUWID('token') parameter in the command. If the thread is indoubt, it must be manually resolved using the RECOVER INDOUBT command. The commit or abort decision made at the coordinator must be determined so that the same decision can be made at DB2.

Contact the database administrator at the coordinator location to determine whether the thread with LUWID=luwid committed or aborted.

If the coordinator system is another DB2, then the following steps can be taken at the DB2 coordinator to determine the commit or abort decision.

1. If the coordinator DB2 system is running and a DB2 conditional restart was not performed since the error was detected at this (the participant) location, ask the database administrator at the DB2 coordinator location to issue one of the following DISPLAY THREAD TYPE(INDOUBT) commands:
   - A -DISPLAY THREAD TYPE(INDOUBT), using the LOCATION keyword and specifying the location name of the local DB2.
   - A -DISPLAY THREAD TYPE(INDOUBT) LUWID(luwid)

   If a thread with LUWID=luwid appears in the display, then the display also includes the commit or abort decision. If no thread with LUWID=luwid appears in the display, then the thread was aborted.
2. If the DB2 coordinator is down or if a conditional restart was performed there, ask the database administrator at the DB2 coordinator to run DSN1LOGP, specifying SUMMARY(ONLY) FILTER LUWID(luwid). The output from DSN1LOGP includes the commit or abort decision.

DSNL413I PROTOCOL ERROR DETECTED IN SYNCPOINT REQUEST/RESPONSE FROM PARTICIPANT *location* **LUWID=***luwid=token***. INFORMATION RECORDED IN TRACE RECORD WITH IFCID=208 AND IFCID SEQUENCE NUMBER=***seq-no*

Explanation: DB2 is the coordinator of the logical unit of work identified by 'luwid=token'. A protocol error occurred during the SNA sync point exchange with the participant identified by 'location'. The protocol error fits into one of the following categories:

- An unknown SNA PS header was received.
- An SNA PS header was received in the wrong sequence.
- An invalid LU6.2 verb sequence was used to deliver the SNA PS header.
- A DEALLOCATE TYPE(ABEND_SVC) was received from the participant, indicating that the participant detected a protocol violation in the PS headers transmitted by DB2.

location
The relational database name (location-name) of the participant.

luwid=token
The global logical unit of work id and the local token that identify the thread.

seq-no The IFCID sequence number associated with the trace record. The sequence number is printed in hexadecimal format. If the sequence number is zero, IFCID 208 was not active or it was active with a destination of the resident trace table only. IFCID 208 is a member of statistics class 4, which has a default destination of SMF.

If the protocol error is detected before the commit decision, the unit of work is backed out.

If the protocol error is detected after the commit decision, the outcome of the unit of work is unchanged. In this case, the remote participant might be indoubt.

DB2 does not automatically resolve the indoubt thread with the partner 'location' after a protocol error is detected.

System action: Alert A005 is generated. A trace record whose IFCID=208 is written.

System programmer response: This is an APARable error, probably at the participant, but possibly at the DB2 coordinator. The DB2 documentation required is contained in the statistics class 4 trace record. Determine the destination of statistics class 4. Locate and print the trace record with SEQUENCE NUMBER='seq-no'.

Call the database administrator. The participant might need to manually resolve the indoubt thread.

Database Administrator Action: The DB2 coordinator does not know the status of the thread at the participant. The DB2 coordinator assumes that it is indoubt and maintains knowledge of the indoubt participant until it is purged with the RESET INDOUBT command. If the thread is indoubt at the participant, it must be resolved using manual (heuristic) means. The status of the thread at the local DB2 (the coordinator) can be determined with the DISPLAY THREAD command for type indoubt threads by including the LUWID('token') parameter in the command.

If the 'participant' location is also a DB2, the RECOVER INDOUBT command can be used at the participant to resolve the indoubt thread.

DSNL414E PROTOCOL ERROR DURING INDOUBT THREAD RESOLUTION WITH PARTICIPANT *location* **LUWID=***luwid=token* **INFORMATION RECORDED IN TRACE RECORD WITH IFCID=206 AND IFCID SEQUENCE NUMBER=***seq-no*

Explanation: DB2 has coordinator responsibility for the logical unit of work identified by 'luwid=token'. The DB2 thread associated with the logical unit of work is either committed or aborted. A protocol error occurred during indoubt resolution with the participant identified by 'location'.

The thread at the participant remains indoubt. DB2 rejects further attempts to automatically resolve the indoubt thread.

location
The relational database name (location-name) of the participant.

luwid=token
The global logical unit of work id and the local token that identify the thread.

seq-no The IFCID sequence number associated with the trace record. The sequence number is printed in hexadecimal format. If the sequence number is zero, IFCID 206 was not active or it was active with a destination of the resident trace table only. IFCID 206 is a member of statistics class 4, which has a default destination of SMF.

System action: Alert A003 is generated. A trace record whose IFCID=206 is written.

System programmer response: This is an APARable error, probably at the participant, but possibly at the DB2 coordinator. The DB2 documentation required is contained in the statistics class 4 trace record. Determine the destination of statistics class 4. Locate and print the trace record with SEQUENCE NUMBER='seq-no'.

Call the database administrator. The participant might need to manually resolve the indoubt thread.

Database Administrator Action: The DB2 coordinator maintains knowledge of the indoubt participant until it is purged with the RESET INDOUBT command. If the thread is indoubt at the participant, it must be manually (heuristically) resolved. The status of the thread at the local DB2 (the coordinator) can be determined with the DISPLAY THREAD command for type indoubt threads by including the LUWID('token') parameter in the command.

If the participant is another DB2, the RECOVER INDOUBT command can be used to resolve the indoubt thread.

DSNL415I AUTOMATIC RESOLUTION OF *luwid=token* **WITH COORDINATOR** *location* **RESULTED IN ABORT**

Explanation: The identified indoubt logical unit of work was automatically resolved by communication with the coordinator location. After automatic resolution, this message is issued if message DSNL405 was previously issued to report the thread as indoubt. This message is not issued if message DSNL405 occurs in one restart of DDF or DB2, and automatic resolution occurs in a different restart of DDF or DB2.

luwid=token
The global logical unit of work id and the local token that identify the thread

location
The field contains one of the following:

- The relational database name (location name)
- The VTAM LU name (<luname>)
- The dotted decimal IP address (nnn.nnn.nnn.nnn).

System action: Processing continues normally.

DSNL416I COLD START CONNECTION REJECTED BY PARTNER *location*

Explanation: DB2 attempted to make a cold-start connection with a partner location. The partner rejected this attempted connection. DB2 never rejects cold-start connections by partners.

location
Either the relational database name (location-name) or the logical unit name (<luname>) of the partner.

System action: The connection was not made.

System programmer response: Call the database administrator.

Database Administrator Action: DB2 is not able to connect the partner 'location' until the partner 'location' allows a cold-start connection with DB2. Contact the database administrator at the partner 'location'.

Contact your IBM Support Center for further assistance.

DSNL417E *csect-name* **COLD START DETECTED AT COORDINATOR LOCATION** *location* **FOR INDOUBT THREAD WITH** *luwid=token***. MANUAL RESOLUTION IS REQUIRED. INFORMATION RECORDED IN TRACE RECORD WITH IFCID=204 and IFC SEQUENCE NUMBER=***seq-no***.**

Explanation: DB2 has resync responsibility for the indoubt thread identified. The partner location informed DB2 that it had performed a cold start operation and lost all knowledge of this indoubt thread. This indoubt thread must be manually resolved with the RECOVER INDOUBT command, followed by the RESET INDOUBT command.

location
The field contains one of the following:
- The relational database name (location name)
- The VTAM LU name (<luname>)
- The dotted decimal IP address (nnn.nnn.nnn.nnn).

luwid=token
The logical unit of work id and the local token that identify the thread.

seq-no The IFCID sequence number associated with the trace record. The sequence number is printed in hexadecimal format. If the sequence number is zero, then IFCID 204 was not active or it was active with a destination of the resident trace table only. IFCID 204 is a member of statistics class 4, which has a default destination of SMF and contains information about a partner's cold start.

System action: Alert A001 is generated. A trace record whose IFCID=204 is written.

System programmer response: Call the database administrator.

Database Administrator Action: DB2 is a participant with one or more indoubt threads where the coordinator is 'location'. The DBMS at 'location' performed a cold start. The DB2 participant assumes that the coordinator recovery log was lost or damaged and indoubt thread resolution cannot be achieved. There may be inconsistent data at the coordinator. A list of indoubt threads involving the coordinator 'location' can be displayed using the DISPLAY THREAD command for indoubt type threads by including the LOCATION('location') parameter in the command.

The heuristic decision (that is, to commit or abort the logical unit of work) should be coordinated with any other participants. The existence of other participants might not be easy to determine. The information might be available in the coordinators recovery log even though the coordinator performed a cold start. If the coordinator is a DB2, then it might be possible to execute the DSN1LOGP utility to produce a summary report. If this is possible, the following information might be produced:
- The LUWID of the thread.
- The NID used by IMS or CICS to identify the thread.
- The name of all participants.
- The name of an upstream coordinator if the coordinator is also a participant. This could happen when an application requester accesses a DB2 AS using application-directed access, which in turn accesses another DB2 server with system-directed access. (The local DB2 might be the third system in this scenario.)

At this DB2 participant, all indoubt threads coordinated by the cold starting location must be resolved with the RECOVER INDOUBT command. Locked data remains unavailable until resolution. Threads that were already resolved with this command are discarded. Threads not yet resolved are discarded after resolution with the command.

The commit or abort decision provided using the RECOVER INDOUBT command for a logical unit of work are propagated to all downstream participants, if any.

DSNL418E *csect-name* **COLD START DETECTED AT PARTICIPANT LOCATION** *location* **FOR INDOUBT THREAD WITH** *luwid=token***. POSSIBLE DAMAGE. INFORMATION RECORDED IN TRACE RECORD WITH IFCID=204 and IFC SEQUENCE NUMBER=***seq-no***.**

Explanation: DB2 has coordinator responsibility for

the indoubt thread identified. The partner location informed DB2 that it had performed a cold start operation and lost all knowledge of this indoubt thread. There might be damage at the participant.

location
The field contains one of the following:
- The relational database name (location name)
- The VTAM LU name (<luname>)
- The dotted decimal IP address (nnn.nnn.nnn.nnn).

luwid=token
The global logical unit of work id and the local token that identify the thread.

seq-no The IFCID sequence number associated with the trace record. The sequence number is printed in hexadecimal format. If the sequence number is zero, then IFCID 204 was not active or it was active with a destination of the resident trace table only. IFCID 204 is a member of statistics class 4, which has a default destination of SMF.

System action: Alert A001 is generated. A trace record whose IFCID=204 is written.

System programmer response: Call the database administrator.

Database Administrator Action: DB2 is the coordinator with indoubt thread resolution responsibility for one or more indoubt logical units of work at 'location'. The DBMS at 'location' performed a cold start. DB2 assumes that the participant recovery log was lost or damaged and indoubt thread resolution cannot be achieved. There is probably inconsistent data at the participant. Minimally, the participant might not completely reflect the final outcome of the threads that were indoubt at the time the failure occurred. A list of indoubt threads involving the partner 'location' can be displayed using the DISPLAY THREAD command for type indoubt threads by including the LOCATION('location) parameter in the command.

Information about the indoubt participant continues to appear in the DISPLAY THREAD TYPE(INDOUBT) report until it is purged with the RESET INDOUBT command.

DSNL419I PARTNER *location* HAS RESOLVED WARM START STATUS AND CONNECTION IS ACCEPTED

Explanation: A successful warm start connection was made with a partner. A previous attempt to make a warm start connection with the partner was rejected with a DSNL410I message because the partner specified an incorrect log name.

location
Either the relational database name (location-name) or the logical unit name (<luname>) of the partner

System action: The connection was successful. Normal processing continues.

DSNL420I COLD START CONNECTION BY PARTICIPANT *location* ACCEPTED. POSSIBLE DAMAGE. INFORMATION RECORDED IN TRACE RECORD WITH IFCID=204 AND IFCID SEQUENCE NUMBER=*seq-no*

Explanation: DB2 has coordinator responsibility for indoubt threads at a participant and just connected with the participant, which lost all knowledge of indoubt threads because of a previous cold start. There might be damage at the participant.

location
The field contains one of the following:
- The relational database name (location name)
- The VTAM LU name (<luname>)
- The dotted decimal IP address (nnn.nnn.nnn.nnn).

seq-no The IFCID sequence number associated with the trace record. The sequence number is printed in hexadecimal format. If the sequence number is zero, then IFCID 204 was not active or it was active with a destination of the resident trace table only. IFCID 204 is a member of statistics class 4, which has a default destination of SMF.

System action: The connection with the partner is accepted. Alert A001 is generated. A trace record whose IFCID=204 is written.

System programmer response: Call the database administrator.

Database Administrator Action: DB2 is the coordinator with indoubt thread resolution responsibility for one or more indoubt logical units of work at 'location'. The DBMS at 'location' performed a cold start. DB2 assumes that the participant recovery log was lost or damaged and indoubt thread resolution cannot be achieved. There is probably inconsistent data at the participant. Minimally, the participant might not completely reflect the final outcome of the threads that were indoubt at the time the failure occurred. A list of indoubt threads involving the partner 'location' can be displayed using the DISPLAY THREAD command for type indoubt threads by including the LOCATION('location) parameter in the command.

Information about the indoubt participant continues to appear in the DISPLAY THREAD TYPE(INDOUBT) report until it is purged with the RESET INDOUBT command.

DSNL421I SNA XLN PROTOCOL VIOLATION BY PARTNER *location*. INFORMATION RECORDED IN TRACE RECORD WITH IFCID=0236 AND IFCID SEQUENCE NUMBER=*seq-no*

Explanation: DB2 detected a protocol violation in the SNA Exchange Log Names (XLN) exchange with the partner at the specified location.

location
Either the relational database name (location-name) or the logical unit name (<luname>) of the partner that sent the invalid XLN message.

seq-no The IFCID sequence number associated with the trace record. The sequence number is printed in hexadecimal format. If the sequence number is zero, then IFCID 0236 was not active or it was active with a destination of the resident trace table only. IFCID 0236 is a member of statistics class 4, which has a default destination of SMF.

System action: The attempt to connect with the remote site fails. DB2 writes a trace record with IFCID 0236.

Operator response: Call the system programmer.

System programmer response: Contact the system programmer for the remote site. The invalid XLN message is recorded in the IFCID 0236 trace record. The system logic error that causes the invalid XLN message must be corrected at the remote site.

DSNL422I WARM START CONNECTION BY PARTNER *location* REJECTED BECAUSE THE PARTNER REMEMBERS OUR LOG NAME INCORRECTLY. OUR LOG NAME IS *name1* AND THE PARTNER REMEMBERS IT AS *name2*

Explanation: An attempt to make a warm start connection with a partner was rejected because the partner specified our log name as name2. Our log name is name1, which is the LUNAME of the local DB2. This error might be a recoverable error if the partner can restart with our log name as name1. If this is not possible, then the partner must perform a cold start connection with DB2.

location
The 'location' with which the connection failed: either the relational database name (location-name) or the logical unit name (<luname>) of the partner.

name1 Our log name, as remembered by us.

name2 Our log name, as remembered by our partner and sent to us in the attempted warm start connection.

System action: The connection with the partner is rejected. DB2 retains indoubt knowledge.

System programmer response: Call the database administrator.

Database Administrator Action: Contact the database administrator at the partner 'location' and determine if it is possible for the partner to perform a warm start with our log name specified as the LUNAME of this DB2. If this is possible, the next attempt to connect will succeed.

If this cannot be done, then there are two other possible solutions.

- Have the partner 'location' perform a cold start connection with this DB2 subsystem.
- Use the RESET INDOUBT command with the FORCE option in conjunction with the LOCATION parameter to cause the next connection with the partner to be a cold start connection.

DSNL423I AUTOMATIC RESOLUTION OF *luwid=token* WITH PARTICIPANT *location* RESULTED IN COMMIT

Explanation: The identified indoubt logical unit of work was automatically resolved through communication with the participant location. After automatic resolution, this message is issued if message DSNL406 was previously issued reporting the thread as indoubt. This message is not issued if message DSNL406 occurs in one restart of DDF or DB2, and automatic resolution occurs in a different restart of DDF or DB2.

luwid=token
The global logical unit of work id and the local token that identify the thread.

location
The field contains one of the following:

- The relational database name (location name)
- The VTAM LU name (<luname>)
- The dotted decimal IP address (nnn.nnn.nnn.nnn).

System action: Processing continues normally.

DSNL424I AUTOMATIC RESOLUTION OF *luwid=token* WITH PARTICIPANT *location* RESULTED IN ABORT

Explanation: The identified indoubt logical unit of work was automatically resolved by communication with the participant location. After automatic resolution, this message is issued if message DSNL425 was previously issued reporting the thread as being indoubt. This message is not issued if message DSNL425 occurs in one restart of DDF or DB2, and

automatic resolution occurs in a different restart of DDF or DB2.

luwid=token
The global logical unit of work id and the local token that identify the thread

location
The field contains one of the following:
- The relational database name (location name)
- The VTAM LU name (<luname>)
- The dotted decimal IP address (nnn.nnn.nnn.nnn).

System action: Processing continues normally.

DSNL425I THREAD *luwid=token* **AT PARTICIPANT** *location* **MAY BE INDOUBT BECAUSE OF A COMMUNICATION FAILURE. DB2 ABORTED THE THREAD. INFORMATION RECORDED IN TRACE RECORD WITH IFCID=209 AND IFCID SEQUENCE NUMBER=***seq-no*

Explanation: During execution of the two phase commit protocol with the participant identified by 'location', communications failed. Phase 1 of the protocol completed with the participant and it is prepared for either commit or abort. The decision to abort the thread was made, but cannot be communicated to the participant at this time. The participant is indoubt.

DB2 becomes responsible for indoubt resolution with the participant. This responsibility appears in a DISPLAY THREAD TYPE (INDOUBT) report. Periodic attempts are made to reestablish communication with the participant for automatic indoubt resolution. If successful, message DSNL424 is written.

luwid=token
The global logical unit of work id and the local token that identify the thread.

location
The field contains one of the following:
- The relational database name (location name)
- The VTAM LU name (<luname>)
- The dotted decimal IP address (nnn.nnn.nnn.nnn).

seq-no The IFCID sequence number associated with the trace record. The sequence number is printed in hexadecimal format. If the sequence number is zero, IFCID 209 was not active or it was active with a destination of the resident trace table only. IFCID 209 is a member of statistics class 4, which has a default destination of SMF.

System action: Alert A006 is generated. A trace record whose IFCID=209 is written.

Operator response: If message DSNL424 for the LUWID specified in message DSNL406 does not appear in a reasonable amount of time, call the system programmer.

System programmer response: Determine the cause of the communication failure and have the problem fixed. DB2 periodically attempts to reestablish communication. If the problem cannot be fixed in a reasonable amount of time, call the database administrator. The participant might need to resolve the indoubt state manually in order to release locked resources.

Database Administrator Action: If the unit of work with LUWID=luwid is indoubt at 'location', inform the database administrator there that the logical unit of work with LUWID=luwid was aborted to assist in manual resolution of the indoubt thread at 'location'.

DSNL429I INCONSISTENT DDF LOG DATA, THE FIRST CONNECTION WITH EACH REMOTE LOCATION WILL BE A COLD START CONNECTION

Explanation: DB2 detected an inconsistency in the DB2 log during restart operations. Since the log is inconsistent and cannot be trusted, DB2 performs a cold start connection with each partner LU the next time a connection with that LU is attempted. If DB2 has indoubt resolution responsibility either as a coordinator or as a participant, then this information is lost as indicated by messages DSNL438 and/or DSNL439, which might follow this message. This means that:
- All information that DB2 maintained about having coordinator responsibility for remote participants was purged. These remote participants need manual resolution for any indoubt threads for which DB2 was the coordinator. A DSNL439I message is displayed for each indoubt remote participant.
- All information that DB2 maintained about having participant responsibility for remote coordinators was purged. A DSNL438I message is displayed for each database access thread that has a remote coordinator. If a database access thread remains indoubt at the end of restart, then the thread must be manually resolved. A -DISPLAY THREAD (*) TYPE(INDOUBT) LOCATION (*) command provides a report that lists the indoubt threads that need manual resolution.

System action: Processing continues normally.

System programmer response: Call the database administrator.

Database Administrator Action: See the DSNL438I and DSNL439I messages to determine what action to take.

DSNL432I DDF IS COORDINATOR FOR *nnnn* THREADS WITH *mmmm* INDOUBT REMOTE PARTNERS

Explanation: This message is issued when DDF is started or stopped, if DDF determines that this DB2 has coordinator responsibility for one or more indoubt threads with one or more partners. This is a summary status message.

nnnn The number of indoubt threads for which DB2 is the coordinator

mmmm
The number of remote partners where the threads are indoubt

System action: Processing continues normally.

Operator response: If the message was issued as a result of DDF starting, no action is necessary. If the message was issued as a result of DDF stopping, contact the database administrator.

Database Administrator Action: If the message was issued as a result of stopping DDF, then DB2 maintained all information necessary to perform automatic resynchronization with partner locations and resolve all indoubt units of work. Issue the START DDF command to initiate this process.

If the message was issued as a result of starting DDF, then DDF started automatic resynchronization.

DSNL433I DDF IS PARTICIPANT FOR *nnnn* INDOUBT THREADS WITH *mmmm* REMOTE PARTNERS

Explanation: This message is issued when DDF is started or stopped, if DDF determines that this DB2 has participant responsibility for one or more indoubt threads with one or more partners. This is a summary status message.

nnnn The number of indoubt threads for which DB2 is the participant

mmmm
The number of remote partners that have coordinator responsibility

System action: Processing continues normally.

Operator response: If the message was issued as a result of DDF starting, no action is necessary. If the message was issued as a result of DDF stopping, contact the database administrator.

Database Administrator Action: If the message was issued as a result of stopping DDF, then DB2 maintained all information necessary to perform automatic resynchronization with partner locations and resolve all indoubt units of work. Issue the START DDF command to initiate this process.

If you do not want to perform automatic resynchronization, you can still resolve the indoubt units of work for which DB2 is a participant and which hold database resources locked. Use the DISPLAY THREAD command for type indoubt threads to produce a display that lists these threads. Then use the RECOVER INDOUBT command to either commit or abort these units of work.

To avoid data inconsistency, coordinate this manual resolution with the database administrators at the locations that have coordinator responsibility for each indoubt unit of work. If any of the indoubt units of work at DB2 also have coordinator responsibility for participant locations, then you should notify the database administrators at the participant locations of the commit or abort decision so they can make the same decision.

DSNL434E DDF NOT STARTED - INCORRECT VTAM RELEASE LEVEL

Explanation: The local VTAM release level is older than the minimum release level required by DB2. When DB2 issues the VTAM OPEN macro, the VTAM release level is returned to DB2. DB2 examines the VTAM release level to determine whether the local VTAM system provides the function required by DB2.

System action: The attempt to START DDF fails.

Operator response: Call the system programmer.

System programmer response: Install a release of VTAM that is compatible with the DB2 system. The DB2 Program Directory identifies the required VTAM release level.

DSNL438I RESYNC STATUS PURGED FOR COORDINATOR *location*. LUWID=*luwid*, URID=*urid*, STATUS=*s* INFORMATION RECORDED IN TRACE RECORD WITH IFCID=235 and IFCID SEQUENCE NUMBER=*seq-no*

Explanation: The information related to resynchronization with the remote coordinator location for the logical unit of work identified by 'luwid', which is displayed in this message, is being purged by DB2. Automatic indoubt resolution responsibility for this distributed logical unit of work can no longer be performed.

This situation can arise in one of two ways;
- A DB2 conditional restart (other than a DB2 cold start; that is, a DB2 conditional restart with STARTRBA=ENDRBA) is being performed.
- An inconsistency in the DB2 log was detected during DB2 restart processing (indicated by a DSNL429I message).

location
The field contains one of the following:

- The relational database name (location name)
- The VTAM LU name (<luname>)
- The dotted decimal IP address (nnn.nnn.nnn.nnn).

luwid The global logical unit of work id that identifies the logical unit of work.

urid The unit of recovery identifier. This is the RBA on the DB2 recovery log of the log record that marks the beginning of the logical unit of work. This is the earliest RBA required to process the logical unit of work during restart.

s Restart status of the logical unit of work. When DB2 came down, the logical unit of work was in one of these states:

A INABORT: The logical unit of work was in the "must-complete" phase of abort, but abort has not yet completed.

C INCOMMIT: The logical unit of work was in the "must-complete" phase of commit, but commit has not yet completed.

D INDOUBT: The logical unit of work completed the first phase of commit, but DB2 has not yet received the commit or abort decision from the coordinator.

F INFLIGHT: The logical unit of work has not yet completed the first phase of commit and should be backed out.

U UNKNOWN: The status of the logical unit of work is not known to DB2 because of conditional restart.

seq-no The IFCID sequence number associated with the trace record. The trace record is written to statistics class 4, which is started to SMF as a default during DB2 installation. Statistics class 4 must be started to a permanent destination (SMF or GTF) for the trace record to be accessible. The sequence number is printed in hexadecimal.

System action: Processing continues normally.

System programmer response: Call the database administrator.

Database Administrator Action: The resynchronization information that was purged by DB2 is recorded in this message. To determine the proper commit or abort decision made by the coordinator at 'location', use the status code as follows:

A The correct decision is to abort the logical unit of work.

C The correct decision is to commit the logical unit of work.

D Contact the database administrator at 'location' to determine the correct decision.

F The correct decision is to abort the logical unit of work.

U Contact the database administrator at 'location' to determine the correct decision.

DSNL439I RESYNC STATUS PURGED FOR PARTICIPANT *location*. LUWID=*luwid*, URID=*urid*, STATUS=*s* INFORMATION RECORDED IN TRACE RECORD WITH IFCID=235 and IFCID SEQUENCE NUMBER=*seq-no*

Explanation: DB2 is purging the information related to resynchronization with the remote participant location for the logical unit of work identified by 'luwid', which is displayed in this message. Automatic indoubt resolution responsibility for this distributed logical unit of work can no longer be performed.

This situation can arise in one of two ways:

- A DB2 conditional restart (other than a DB2 cold start; that is, a DB2 conditional restart with STARTRBA=ENDRBA) is being performed.
- An inconsistency in the DB2 log was detected during DB2 restart processing (indicated by a DSNL429I message).

location
The field contains one of the following:

- The relational database name (location name)
- The VTAM LU name (<luname>)
- The dotted decimal IP address (nnn.nnn.nnn.nnn).

luwid The global logical unit of work id that identifies the logical unit of work.

urid The unit of recovery identifier. This is the RBA on the DB2 recovery log of the log record that marks the beginning of the logical unit of work. This is the earliest RBA required to process the logical unit of work during restart.

s Restart status of the logical unit of work. When DB2 came down, the logical unit of work was in one of these states:

A INABORT: The logical unit of work was in the "must-complete" phase of abort, but abort has not yet completed.

C INCOMMIT: The logical unit of work was in the "must-complete" phase of commit, but commit has not yet completed.

D INDOUBT: The logical unit of work completed the first phase of commit, but DB2 has not yet received the commit or abort decision from the coordinator.

F INFLIGHT: The logical unit of work has not yet completed the first phase of commit and should be backed out.

U UNKNOWN: The status of the logical unit of work is not known to DB2 because of conditional restart.

seq-no The IFCID sequence number associated with the trace record. The trace record is written to statistics class 4, which is started to SMF as a default during DB2 installation. Statistics class 4 must be started to a permanent destination (SMF or GTF) for the trace record to be accessible. The sequence number is printed in hexadecimal.

System action: Processing continues normally.

System programmer response: Call the database administrator.

Database Administrator Action: The resynchronization information that was purged by DB2 is recorded in this message. When communicating the proper commit or abort decision to the database administrator at 'location', use the status code as follows:

A Abort the logical unit of work.

C Commit the logical unit of work.

D See the DSNL438I message produced for the same 'luwid' and 'urid' to determine the coordinator 'location'. Contact the database administrator at 'location' to determine the correct decision.

F Abort the logical unit of work.

U If there is a DSNL438I message for the same 'luwid' and 'urid', proceed as described above under 'D'. Otherwise, there might be information available on the DB2 log that a DSN1LOGP summary report can provide. If this is not successful, contact the IBM Support Center.

DSNL440I *csect-name* **RESET INDOUBT COMMAND SPECIFICATION IN ERROR**

Explanation: A RESET INDOUBT command was issued, but the specification of keywords or keyword parameter values was in error.

System action: No indoubt information was purged. System processing continues normally.

User response: One of the following errors was made in the RESET INDOUBT command specification:

- RESET INDOUBT was issued without any other keywords specified. At least one of the following keywords must also be specified: LUNAME, IPADDR, LOCATION, or LUWID.
- RESET INDOUBT was issued with both the LUWID and LOCATION keywords specified, but a list of LOCATION keyword parameter values was specified. When both the LUWID and LOCATION keywords are specified only one LOCATION keyword parameter value is allowed.
- RESET INDOUBT was issued with at least one LOCATION keyword parameter value that began with a '<', but did not end with a '>', or the length of the bracketed value was not 1 to 8 characters. When the LOCATION keyword parameter value is specified in the <luname> format, the luname value must be bracketed with '<' and '>' and be 1 to 8 characters long.
- RESET INDOUBT was issued with an invalid IPADDR keyword parameter value. Verify that the specified IPADDRs are in the proper dotted decimal form, that there is a colon delimeter between the IPADDR and port, and that the port number is valid.

If needed, issue the DISPLAY THREAD TYPE(INDOUBT) command to display information (including LOCATION, LUWID, and LUNAME values) about all indoubt threads. Correct the RESET INDOUBT command specification, and reissue the command.

DSNL441I KEYWORD 'LUWID' PARAMETER VALUE *parmvalue* **FORMAT IN ERROR**

Explanation: A RESET INDOUBT command with the LUWID parameter was issued, but the specified parameter value (parmvalue) has an invalid format for an LUWID. There are two correct formats for the LUWID parameter. The first correct format is called a token, and is a 1 to 5 digit decimal number. The second correct format is 'netid.luname.luwunique.comseq', where:

netid A 1 to 8 character network ID

luname
A 1 to 8 character luname

luwunique
A 12 character hexadecimal number

comseq
A 4 character hexadecimal number

All fields in the second correct format must be separated by a period (.) delimiter.

System action: Indoubt information about the specified logical unit of work is not purged from the system.

User response: To determine the correct LUWID value for the unit of work indoubt information that you want

to purge, issue the DISPLAY THREAD TYPE(INDOUBT) command to display all units of work for which indoubt information exists. Then reissue the RESET INDOUBT command, specifying the correct LUWID parameter value.

DSNL442I QUALIFYING INDOUBT INFORMATION FOR LOCATION *location* NOT FOUND

Explanation: A -RESET INDOUBT LOCATION('location') command was issued, but DB2 has no indoubt information about the specified location.

location
The value specified after the LOCATION keyword in the RESET INDOUBT command.

System action: No indoubt information was purged. System processing continues normally.

User response: Use the -DISPLAY THREAD(*) TYPE(INDOUBT) LOCATION(*) command to list the locations for which DB2 has indoubt information. After determining the correct location, reissue the command.

DSNL443I QUALIFYING INDOUBT INFORMATION FOR LUWID=*value* NOT FOUND

Explanation: A -RESET INDOUBT LUWID('value') command was issued where 'value' is either an luwid or the local token that represents an luwid. Either DB2 has no indoubt information about the specified luwid or token, or DB2 has indoubt information that cannot be purged using the -RESET INDOUBT command with the LUWID option.

value The global logical unit of work id or the local token that was specified for the LUWID keyword in the RESET INDOUBT command.

System action: No indoubt information was purged. System processing continues normally.

User response: Use the -DISPLAY THREAD(*) TYPE(INDOUBT) LOCATION(*) command to list the indoubt information known to DB2. If the -DISPLAY THREAD output does not include information about the specified luwid, then that is the reason the message was issued. If the -DISPLAY THREAD output does include information about the specified luwid, this information cannot be purged using the -RESET INDOUBT command with the LUWID option. Refer to the description of each message produced by the -DISPLAY THREAD output that relates to the specified luwid in order to determine how to proceed. After determining the correct luwid or token, reissue the command.

DSNL444I QUALIFYING INDOUBT INFORMATION FOR LOCATION *location* HAS BEEN PURGED

Explanation: A -RESET INDOUBT LOCATION('location') command issued to purge indoubt information about the named location completed successfully and the indoubt information was purged.

location
Either the relational database name (location-name) or the logical unit name (<luname>) that was specified after the LOCATION keyword on the RESET INDOUBT command.

System action: The specified indoubt information was purged. System processing continues normally.

DSNL445I QUALIFYING INDOUBT INFORMATION FOR LUWID *value* HAS BEEN PURGED

Explanation: A -RESET INDOUBT LUWID('value') command, which was issued to purge indoubt information about the specified luwid, completed successfully. The indoubt information was purged.

value The value specified for the LUWID keyword in the -RESET INDOUBT LUWID(value) command that is being processed.

System action: System processing continues normally.

DSNL446I COLD START WILL BE PERFORMED WITH *nettype netvalue*

Explanation: A -RESET INDOUBT LUNAME('luname') FORCE or a -RESET INDOUBT IPADDR('ipaddr') FORCE command issued to force a cold start completed successfully. The next connection with 'luname' or 'ipaddr' is a cold start.

nettype
Identifies the type of network used to access the remote site. There are two possible values:

LU -- indicates that VTAM is used to connect to the remote site.

IPADDR --
indicates that TCP/IP is used to connect to the remote site.

netvalue
The network address of the remote location.

- If *nettype* is LU, *netvalue* specifies the VTAM LU name of the remote location.
- If *nettype* is IPADDR, *netvalue* specifies the dotted decimal IP address of the remote location.

System action: System processing continues normally.

DSNL447I CANNOT DELETE ENTRY *'value'* BECAUSE THERE IS AN INDOUBT THREAD TO BE RESOLVED FIRST

Explanation: A -RESET INDOUBT command was issued with LUNAME(value), IPADDR(value), LOCATION(value), or LUWID(value). The 'value' is an LU name, an IP address, a location, ora LUWID (or its token), that identifies a logical unit of work for which DB2 has participant indoubt resolution responsibility, coordinator indoubt resolution responsibility, or both.

System action: Resynchronization responsibility information was not purged. System processing continues normally.

User response: Use the information in this section to determine the correct action to take. Then issue the proper command to perform that action.

The two most probable causes for the problem are that the -RESET INDOUBT command was issued with the wrong parameters, or that an incorrect 'value' was specified. Use the -DISPLAY THREAD(*) TYPE(INDOUBT) LOCATION(*) command to display the indoubt information known to DB2.

1. If the intent was to purge participant or coordinator indoubt resolution responsibility, then examine the -DISPLAY THREAD output to determine the location name, or names, for which this responsibility is to be purged. Then use the -RESET INDOUBT LUNAME(luname) command with the FORCE option or the -RESET INDOUBT IPADDR(ipaddr) command with the FORCE option to purge this responsibility.

   To determine the luname or IP address and resync port for an upstream coordinator that can be used with the FORCE option, look at the coordinator information from the DSNV408I message in the -DISPLAY THREAD output. There are two tokens associated with the coordinator information that are delimited by a colon (":").

   - The first token is the partner location name or "<luname>" for an SNA connection, or the dotted decimal IP address for a TCP/IP connection (nnn.nnn.nnn.nnn).
   - The second token is the partner LU name for an SNA connection, or the resync port number for a TCP/IP connection.

   To determine the luname or IP address and resync port for downstream participants that can be used with the FORCE option, look at the location names from the DSNV450I and DSNV446I messages. There are two or three tokens associated with the participant location which are delimited by a colon (":").

   - The first token is the partner location name.
   - The second token is the partner LU name for an SNA connection, or the dotted decimal IP address for a TCP/IP connection (nnn.nnn.nnn.nnn).
   - The third token only appears if the second token is an IP address in which case the third token is the partner resync port number.

   See the description of the -RESET INDOUBT command in *DB2 Command Reference* for more information.

   Remember that the FORCE option causes DB2 to purge all knowledge of indoubt resolution responsibility involving the luname or ipaddrspecified on the command.

   Use caution when using the FORCE parameter to purge participant indoubt information when no errors are precluding automatic resolution. Normally, after the use of the RECOVER INDOUBT command, automatic resolution with the coordinator determines if heuristic damage occurred. This detection is lost if RESET INDOUBT is used before automatic resolution with the coordinator can be achieved.

   Use caution when using the FORCE parameter to purge coordinator indoubt information when no errors are precluding automatic resolution. Once the information is purged, any participant that is indoubt is forced to use a heuristic decision process to resolve the indoubt logical unit of work.
2. Examine the -DISPLAY THREAD output. If the value specified in the command (and displayed in the DSNL447I message) is incorrect, determine the correct value and reissue the command using the correct value.

DSNL448I UNABLE TO FORCE COLD START WITH *nettype netvalue*, SESSION STILL EXISTS

Explanation: A -RESET INDOUBT LUNAME('luname') FORCE or a -RESET INDOUBT IPADDR('ipaddr') FORCE command was issued to force a cold start, but network connections still exist between DB2 and the partner 'luname' or 'ipaddr'. The information related to this partner was not purged, so a cold start is not performed.

nettype
Identifies the type of network used to access the remote site. There are two possible values:

LU -- indicates that VTAM is used to connect to the remote site.

IPADDR -- indicates that TCP/IP is used to connect to the remote site.

netvalue
The network address of the remote location.

- If *nettype* is LU, *netvalue* specifies the VTAM LU name of the remote location.

- If *nettype* is IPADDR, *netvalue* specifies the dotted decimal IP address of the remote location.

System action: System processing continues normally.

User response: For SNA, you can use the VTAM command DISPLAY NET to determine the existence of sessions between two logical units. Terminate the sessions by using the VTAM command VARY NET,TERM to allow the RESET INDOUBT to occur. For TCP/IP, terminate all connections to the IP address.Determine the correct luname or ipaddr and reissue the command.

DSNL449I DB2 HAS NO INFORMATION RELATED TO LUNAME *luname*

Explanation: A -RESET INDOUBT LUNAME('luname') FORCE command was issued but DB2 has no knowledge of the specified 'luname'.

luname
The logical unit name, as defined to VTAM, of the coordinator or participant

System action: No information was purged. System processing continues normally.

User response: Determine the correct luname as follows:

- Use the -DISPLAY THREAD(*) TYPE(INDOUBT) LOCATION(*) command to determine locations for which DB2 has indoubt responsibility.
- Query the SYSIBM.LOCATIONS table to correlate a location name with an luname. The LINKNAME column contains the luname associated with the location name in the LOCATION column.

Then reissue the command, specifying the correct luname.

DSNL450I DDF THREAD *luwid=token* **IS BETWEEN PHASE 1 AND PHASE 2 OF COMMIT. THREAD IS NOW AN INDOUBT THREAD AS A RESULT OF CANCEL REQUEST.**

Explanation: The identified DDF thread was in a prepared state waiting for the commit or backout decision of the commit coordinator. The thread cannot be canceled in this state. However, the thread was removed from the active state and placed into the indoubt state. The conversation with the coordinator was deallocated. The thread is now indoubt and appears in the display thread report for indoubt threads. The resources held by the thread continue to be held. Automatic resolution of the indoubt state is attempted periodically. Resources held by the thread are released after indoubt resolution occurs.

luwid=token
The global logical unit of work id and the local token that identify the thread

System action: Processing continues normally.

Operator response: If the purpose of canceling this thread was to free the resources held by the thread, call the database administrator.

Database Administrator Action: To release the resources held by the indoubt thread before automatic resolution with the coordinator, a manual commit or abort decision can be made with the RECOVER INDOUBT command. First, issue the DISPLAY THREAD command for an indoubt thread report, which includes the subject thread. From this report, determine the name of the coordinator. Contact the coordinator system and find out whether a commit or abort decision was made.

DSNL451I QUALIFYING INDOUBT INFORMATION FOR LUNAME *luname* **NOT FOUND**

Explanation: A -RESET INDOUBT LUNAME('luname') command was issued, but DB2 has no indoubt information about the specified luname.

luname
The logical unit name, as defined to VTAM, of the coordinator or participant

System action: No indoubt information was purged. System processing continues normally.

User response: Use the -DISPLAY THREAD(*) TYPE(INDOUBT) LOCATION(*) command to list the locations for which DB2 has indoubt information. Entries associated with a particular LUNAME have a location name in this form: '<luname>'. Use 'luname' (without the enclosing brackets) as the input string to RECOVER INDOUBT.

After determining the correct luname, reissue the command.

DSNL452I QUALIFYING INDOUBT INFORMATION FOR LUNAME *luname* **HAS BEEN PURGED**

Explanation: A -RESET INDOUBT LUNAME('luname') command, which was issued to purge indoubt information about the specified luname completed successfully. The indoubt information was purged.

luname
The luname specified for the LUNAME keyword in the -RESET INDOUBT LUNAME(luname) command being processed.

System action: The specified indoubt information was purged. System processing continues normally.

DSNL453I *csect* **QUALIFYING INDOUBT INFORMATION FOR IPADDR***ipaddr* **NOT FOUND**

Explanation: A -RESET INDOUBT IPADDR('ipaddr') command was issued, but DB2 has no indoubt information about the specified IP address.

csect The CSECT name of the DB2 module that issued the message

ipaddr The dotted decimal IP address of the coordinator or participant

System action: No indoubt information was purged. System processing continues normally.

User response: Use the -DISPLAY THREAD(*) TYPE(INDOUBT) LOCATION(*) command to list the locations for which DB2 has indoubt information. Entries associated with TCP/IP have a string with this format in the LOCATION field: "nnn.nnn.nnn.nnn".

After determining the correct *ipaddr*, reissue the command.

DSNL454I *csect* **QUALIFYING INDOUBT INFORMATION FOR IP ADDRESS** *ipaddr* **HAS BEEN PURGED**

Explanation: A -RESET INDOUBT IPADDR('ipaddr') command, which was issued to purge indoubt information about the specified IP address completed successfully. The indoubt information was purged.

csect The CSECT name of the DB2 module that issued the message

ipaddr The IP address specified for the IPADDR keyword in the -RESET INDOUBT IPADDR(ipaddr) command being processed

System action: The specified indoubt information was purged. System processing continues normally.

DSNL455I *csect* **DB2 HAS NO INFORMATION RELATED TO IP ADDRESS** *ipaddr*

Explanation: A -RESET INDOUBT IPADDR('ipaddr') FORCE command was issued but DB2 has no knowledge of the specified 'ipaddr'.

csect The CSECT name of the DB2 module that issued the message.

ipaddr The dotted decimal TCP/IP address of the coordinator or participant

System action: No information was purged. System processing continues normally.

User response: Determine the correct luname as follows:

- Use the -DISPLAY THREAD(*) TYPE(INDOUBT) LOCATION(*) command to determine locations for which DB2 has indoubt responsibility. Entries associated with TCP/IP locations specify "nnn.nnn.nnn.nnn" in the LOCATION field.

Reissue the command; specify the correct ipaddr.

DSNL500I **CONVERSATION FAILED TO LOCATION=***locname* **RESOURCE=***luname.modename***.** *convid.sessid* **RTNCD=***global vtam primary return code* **FDBK2=***global vtam secondary return code* **RCPRI=***appc-specific primary return code* **RCSEC=***appc-specific secondary return code* **SENSE=***sna sense code*

Explanation: An advanced program-to-program communications (APPC) conversation failed, or could not be allocated, because a VTAM return code indicates a conversation failure. VTAM return codes are defined in *z/OS Communicatons Server SNA Programmer's LU 6.2 Reference*.

SNA sense codes that start with 00 represent DB2 reason codes. Refer to *DB2 Codes* for a description of more common SNA sense codes returned to DB2 by VTAM.

The variables that are not return codes are described as follows:

Variable
Description

locname The location name associated with the remote site.

luname The VTAM LU name associated with the remote location.

modename
The logon mode used by DB2 to allocate the conversation.

convid The conversation identifier given DB2 by VTAM when the conversation was allocated.

sessid The session identifier given DB2 by VTAM when the conversation was allocated.

System action: Communication on the conversation is discontinued.

Operator response: Notify the system programmer.

System programmer response: Refer to *z/OS Communicatons Server SNA Programmer's LU 6.2 Reference* for failure analysis or *DB2 Codes* for information on reason codes.

DSNL501I **CNOS PROCESSING FAILED FOR LU** *luname* **AND MODE** *modename* **RTNCD=***global vtam return code* **FDBK2=***global vtam secondary return code* **RCPRI=***appc-specific primary return code* **RCSEC=***appc-specific secondary return code* **SENSE=***vtam sense code*

Explanation: A failure occurred while attempting to negotiate limits with the specified location for a LU and mode. VTAM return codes are described in *z/OS Communicatons Server SNA Programmer's LU 6.2 Reference*. VTAM sense codes that start with 00 represent DB2 reason codes.

The variables are described as follows:

Variable
Description

luname
The VTAM LU name associated with the remote location.

modename
The logon mode used by DB2 to allocate the conversation.

System action: All communication to the remote location fails until the limits can be negotiated with the remote location.

Operator response: Notify the system programmer.

System programmer response: Refer to *z/OS Communicatons Server SNA Programmer's LU 6.2 Reference* for failure analysis or *DB2 Codes* for information on reason codes.

DSNL502I **SYSTEM CONVERSATION FAILED TO LOCATION** *locname* **RESOURCE=***luname.modename***.** *convid.sessid* **RTNCD=***global vtam primary return code* **FDBK2=***global vtam secondary return code* **RCPRI=***appc-specific primary return code* **RCSEC=***appc-specific secondary return code* **SENSE=***sna sense code*

Explanation: A system conversation failed, or could not be allocated, because VTAM returned a nonzero return code. VTAM return codes are defined in *z/OS Communicatons Server SNA Programmer's LU 6.2 Reference*.

SNA sense codes that start with 00 represent DB2 reason codes. Refer to *DB2 Codes* for a description of more common SNA sense codes returned to DB2 by VTAM.

The variables are described as follows:

Variable
Description

locname The location name associated with the remote site.

luname The VTAM LU name associated with the remote location.

modename
The logon mode used by DB2 to allocate the conversation.

convid The conversation identifier given DB2 by VTAM when the conversation was allocated.

sessid The session identifier given DB2 by VTAM when the conversation was allocated.

System action: Communication on the conversation is not allowed.

Operator response: Notify the system programmer.

System programmer response: Refer to *z/OS Communicatons Server SNA Programmer's LU 6.2 Reference* for failure analysis or *DB2 Codes* for information on reason codes.

DSNL504I *csect-name* **RESOURCE UNAVAILABLE REASON=***reason* **TYPE=***type* **LOCATION=***location* **NAME=***name*

Explanation: The requested function requires a resource that is currently unavailable. The reason code, type, location, and name identifies the resource.

Operator response: Note the reason, type codes, location, and name.

Problem determination: The reason for the resource's unavailability is identified by the reason code. The reason codes that appear in this message are described in *DB2 Codes*. Refer to Table 2 in Appendix A, "Problem determination," on page 659 for an explanation of resource type codes.

DSNL510I *csect-name* **CONVLIMIT NEGOTIATED FROM** *x* **TO** *y* **WITH LOCATION** *locname* **FOR LU** *luname* **AND MODE** *modename*

Explanation: The maximum conversation limit is changed for a mode with a remote subsystem. The limit might change when a CNOS request is received from or sent to a remote subsystem, and the limit is determined by VTAM.

x The most recent value of CONVLIMIT before the last CNOS request was processed by advanced program-to-program communications (APPC)/VTAM for the specified mode name to the specified location.

y The current value of CONVLIMIT for the specified mode name to the specified location.

locname
The location name associated with the remote site that sent the CNOS request to the local DB2 subsystem, or received the CNOS request from the local DB2 subsystem.

luname
The VTAM LU name associated with the remote location that had its CONVLIMIT changed.

modename
The logon mode for the specified location that had its CONVLIMIT changed.

System action: Conversation requests are queued when the maximum conversation limit is reached for a mode to remote subsystem.

DSNL511I *csect* **TCP/IP CONVERSATION FAILED TO LOCATION** *locname* **IPADDR=***ipaddr* **PORT=***port* **SOCKET=***socket* **RETURN CODE=***'rc'***X REASON CODE=***'reason'***X**

Explanation: A TCP/IP socket call failed with the error condition identified by *rc* and *reason*.

Variable
Description

csect The CSECT name of the DB2 module that issued the message.

locname
The location name associated with the remote site.

ipaddr The IP address of the remote site in dotted decimal format.

port The TCP/IP port used by the remote site.

socket The TCP/IP socket call that returned the error.

rc The error condition reported by TCP/IP. The
return code values are documented in *z/OS*
UNIX System Services Messages and Codes.

reason Provides further detail associated with the
TCP/IP return code. The reason code values
are documented in *z/OS UNIX System Services*
Messages and Codes.

System action: Communication on the conversation is discontinued.

User response: Notify the system programmer.

Operator response: Notify the system programmer.

System programmer response: Refer to *z/OS UNIX*
System Services Messages and Codes for an explanation of
the error.

Problem determination: See System Programmer Response.

DSNL512I *csect* **TCP/IP** *socket* **FAILED WITH RETURN CODE=***'rc'***X AND REASON CODE=***'reason'***X**

Explanation: A TCP/IP socket call used to establish communication between DB2 and TCP/IP failed. The following list describes what each of the variables mean:

csect The CSECT name of the DB2 module that issued the message.

socket The name of the TCP/IP socket call that failed.

rc The return code reported by TCP/IP. The
return code values are documented in *z/OS*
UNIX System Services Messages and Codes.

reason The reason code reported by TCP/IP. The
reason code values are documented in *z/OS*
UNIX System Services Messages and Codes.

System action: DB2 may periodically retry the socket call to establish TCP/IP communication.

User response: Notify the system programmer.

Operator response: Notify the system programmer.

System programmer response: Refer to *z/OS UNIX*
System Services Messages and Codes for an explanation of
the error.

| • If *socket*=POLL, then the online ZPARM was changed
| using the SET SYSPARM command, changing the
| MAXDBAT from a positive value to zero. In this
| case, the message is informational, indicating that
| DB2 terminated its TCP/IP listening capability for
| SQL related work because DB2 can no longer accept
| SQL related connections from remote partners when
| MAXDBAT is set to zero.

| • If *socket*=SHUTDOWN_REG (USS_UNAVAILABLE),
| then the UNIX System Services (USS/OMVS)
| environment is not available. DDF start processing
| continues and this message is periodically repeated
| until the Unix System Services environment becomes
| available or DDF is stopped. You can use the z/OS
| "F OMVS, RESTART" command to start the Unix
| System Services environment.

• If *socket*=SETRLIMIT and *reason code*=1148033C, then
you should define DB2 as a superuser. See *DB2*
Installation Guide for instructions on enabling DDF as
a superuser.

• If *socket*=TCPKEEPALIVE, then DB2 issues this
socket call as a result of the TCP/IP KeepAlive value
specified in the DB2 parameters module (usually
DSNZPARM). You may specify this value by the
TCP/IP KEEPALIVE value in the DSNTIP5
installation panel. Another way to specify the value
is through the TCPKPALV value in the DSN6FAC
macro, used to build the DB2 parameters module.

In this situation, processing continues, and DB2 does
not retry the socket call. Verify that the TCP/IP stack
being used by DB2 contains the required companion
support for this function. If not, modify the DB2
parameter module to prevent DB2 from attempting
to override the TCP/IP stack value.

• If *socket*=GETHOSTBYADDR(ip-address), then DB2
invoked a TCP/IP GetHostByAddr socket call to
determine the domain name for the specified IP
address. However, the call failed, and DB2 is not able
to determine the domain name. DB2 retries the
socket call periodically.

During DDF initialization, DB2 first determines the
IP address via a GetHostID socket call. The IP

address returned is the first IP address listed in the
HOME statement of the TCP/IP profile for the
selected TCP/IP stack. Verify the IP address. DDF
must be stopped and restarted in order for DDF to
obtain a new IP address.

Attention: The IP address returned could depend on
the selected TCP/IP stack, in the event that there are
multiple TCP/IP stacks. DB2 requires z/OS UNIX
System Services sockets, and it is z/OS UNIX System
Services that determines the TCP/IP stack to use.
z/OS UNIX System Services provides a way for
users to select a specific TCP/IP stack by using the
BPXTCAFF utility. For more information on
BPXTCAFF, see *z/OS UNIX System Services Planning*.

Using this IP address, DB2 issues a GetHostByAddr
socket call to determine the associated domain name.
The OS/390 Resolver handles the GetHostByAddr
socket call because DB2 uses the z/OS UNIX System
Services Asynchronous Sockets interface. This
resolver uses the resolver configuration information
available to DDF. Until this information for the
selected TCP/IP stack is correctly specified for DDF,
the OS/390 Resolver will be unable to determine the
domain name for DDF. You can find the search order
for the "resolver configuration files" for an OS/390
application, such as DDF, in the IP Configuration
manual for the appropriate level of the
Communication Server for your OS/390 operating
system.

Problem determination: See System Programmer Response.

DSNL513I *csect* **TCP/IP GETHOSTBYNAME FAILED WITH ERRNO=***errno* **FOR DOMAIN** *domain*

Explanation: DB2 encountered a TCP/IP gethostbyname socket call error when attempting to convert a TCP/IP domain name into an IP address.

Variable
Description

csect The CSECT name of the DB2 module that issued the message.

errno The error condition reported by TCP/IP. The *errno* values are documented in *IBM TCP/IP for MVS: Application Programming Reference*.

domain
The TCP/IP domain name that DB2 wants to convert into an IP address. DB2 obtains the *domain* value from the IPADDR column of the SYSIBM.IPNAMES communications database table.

System action: The row in SYSIBM.IPNAMES containing the specified *domain* in the IPADDR column is invalid. If the error occurs during DDF startup, the row in SYSIBM.IPNAMES is ignored. If the error occurs during an SQL statement, the SQL statement fails and the location associated with the SYSIBM.IPNAMES row is not available for TCP/IP communication.

User response: Notify the system programmer.

Operator response: Notify the system programmer.

System programmer response: Refer to *IBM TCP/IP for MVS: Application Programming Reference* for an explanation of the error. Correct the IPADDR column value in SYSIBM.IPNAMES or register the *domain* with TCP/IP.

Problem determination: See System Programmer Response.

DSNL514I *csect-name* **TCP/IP SERVICE** *serv* **IS NOT DEFINED**

Explanation: The TCP/IP getservbyname socket call did not find a matching definition for *serv*.

Variable
Description

serv The TCP/IP service name that DB2 wants to convert into a TCP/IP port number. DB2 obtains the service name from the PORT column of the SYSIBM.LOCATIONS communication database table.

System action: The row in SYSIBM.LOCATIONS containing the specified service name in the PORT column is invalid. The associated LOCATION cannot be used for TCP/IP communication. If the error occurs during an SQL statement, the SQL statement fails.

User response: Notify the system programmer.

Operator response: Notify the system programmer.

System programmer response: If the value specified by *serv* is incorrect, correct the PORT column value in SYSIBM.LOCATIONS.

If the value specified by *serv* is correct, register the service name with TCP/IP.

TCP/IP service names are case sensitive, so make sure that the PORT column in SYSIBM.LOCATIONS is an exact match for the service name defined to TCP/IP.

Problem determination: See System Programmer Response.

DSNL515I *csect* **TCP/IP BIND FOR PORT** *port* **FAILED WITH RETURN CODE=***'rc'***X AND REASON CODE=***'reason'***X**

Explanation: DB2 encountered an error on the TCP/IP bind socket call while attempting to register a TCP/IP port number with the TCP/IP instance.

Variable
Description

csect The CSECT name of the DB2 module that issued the message.

port The TCP/IP port number DB2 wants to register with the TCP/IP instance.

rc The return code reported by TCP/IP. The
return code values are documented in *z/OS*
UNIX System Services Messages and Codes.

reason The reason code reported by TCP/IP. The
reason code values are documented in *z/OS*
UNIX System Services Messages and Codes.

System action: If this error occurs for the TCP/IP port that is used for incoming DRDA requests, the error is probably caused by DB2 restarting on a CEC that already has another DB2 subsystem using the DRDA port number. DB2 will continue to try to activate the TCP/IP port number periodically.

| If this error occurs for a TCP/IP port that is used for
| incoming DRDA requests to an ALIAS, the error is
| probably caused by some of the reasons that an error
| would occur for the normal DRDA request port. Check
| to ensure that the port value was also not previously
| reserved for another application in the system's TCP/IP
| profile. DB2 will continue to try to periodically activate
| the TCP/IP port number.

If this error occurs for the DRDA resynchronization port, DB2 invokes internal processing that is equivalent to STOP DDF MODE(FORCE).

User response: Notify the system programmer.

Operator response: Notify the system programmer.

System programmer response: Refer to *z/OS UNIX*
System Services Messages and Codes for an explanation of
the error.

| Define DB2's TCP/IP port number to the TCP/IP
| instance, or use the change log inventory utility to
| correct the TCP/IP port number used by DB2.

Problem determination: See System Programmer Response.

DSNL516I *csect-name* **LANGUAGE ENVIRONMENT SERVICES NOT AVAILABLE. CEEPIPI ROUTINE CANNOT BE LOADED**

Explanation: DB2 must load the Language Environment interface module (CEEPIPI) to process functions requiring Language Environment services. These Language Environment services are required for TCP/IP socket calls, encryption and decryption, and so on. If the CEEPIPI module is not included in an accessible data set such as STEPLIB data sets or the LPA library, DDF is unable to use the Language Environment.

System action: DDF startup continues, but DDF users cannot use DB2 functions that require Language Environment services.

System programmer response: Add the Language Environment library to the DDF STEPLIB concatenation or to the LPA library. The Language Environment library must be APF authorized when it appears in the DDF STEPLIB concatenation.

Problem determination: Collect the following diagnostic item listed in Appendix A, "Problem determination," on page 659: 1.

DSNL517I *csect-name* **CEEPIPI** *func* **SERVICE FAILED WITH RETURN CODE=***rc*

Explanation: DB2 uses the Language Environment CEEPIPI service to issue certain TCP/IP socket calls. The Language Environment return code *rc* was not expected for the indicated Language Environment function call.

func The function requested by DB2. Possible values are:

add_entry(module-name)
Add an entry to the Language Environment PIPI table dynamically. Module-name is the name of the module that failed to be added to the PIPI table.

call_sub
Run a program that issues a TCP/IP socket call using an existing Language Environment run time environment.

del_sub
Delete an entry from the Language Environment PIPI table dynamically.

init_sub
Create Language Environment sub run time environment.

term Terminate the Language Environment run time environment.

rc The return code received from CEEPIPI.

System action: The attempt to issue the Language Environment request fails.

If this message is issued during DDF startup, DDF startup continues.

If this message is issued during an SQL statement, a -904 SQLCODE is returned to the calling application.

System programmer response: See Problem Determination.

Problem determination: Collect the following diagnostic item listed in Appendix A, "Problem determination," on page 659: 1.

Refer to the Language Environment return codes from CEEPIPI in *Language Environment for OS/390 & VM Programming Guide* for additional information.

If the function requested by DB2 was

add-entry(module-name), an incorrect link edit of the module could create unresolved references. To determine if link edit problems exist, run the OS/390 AMBLIST utility with LISTLOAD and OUTPUT=XREF.

DSNL518I *csect-name* **LE/370 ENCLAVE FAILED WITH RETURN CODE=***erc* **AND FEEDBACK=***fi*

Explanation: DB2 uses the Language Environment CEEPIPI call_sub service to execute certain TCP/IP socket calls. The CEEPIPI interface returned information indicating a failure occurred while executing the TCP/IP socket.

erc The Language Environment enclave return code.

fi Contains one or two message identifiers. The message identifiers are constructed from the Language Environment condition token.

The message identifiers are constructed as follows:
- Single message: Facility_Id combined with Msg_No
- Two messages: Facility_Id combined with Class_Code and Facility_Id combined with Cause_Code

System action: The SQL statement fails with SQLCODE -904.

System programmer response: A description of the Language Environment enclave return code and the condition token are documented in *Language Environment for OS/390 & VM Programming Guide*.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1. 4.

DSNL519I *csect* **TCP/IP SERVICES AVAILABLE FOR DOMAIN** *domain* **AND PORT** *tcpport*

Explanation: DB2 is ready to accept connections on any IP address supported by the TCP/IP stack. This message is issued when re-initialization is successful. DB2 will not accept any connection request until this message is issued.

Variable **Description**

csect The CSECT name of the DB2 module that issued the message.

domain The TCP/IP domain name used by this DB2 subsystem to accept incoming connection requests.

tcpport The TCP/IP port number used by this DB2 subsystem to accept incoming connection requests.

System action: The DB2 TCP/IP port number is available to accept incoming connection requests on any IP address supported by the TCP/IP stack.

DSNL520I *csect* **TCP/IP SERVICES AVAILABLE FOR RESPORT** *port*

Explanation: DB2 encountered a TCP/IP failure during DDF startup processing. The DSNL515I message identifies the cause of the failure. After the failure is detected, DDF periodically attempts to reinitialize TCP/IP to allow TCP/IP resynchronization to occur. This message is issued when reinitialization is successful.

Variable **Description**

csect The CSECT name of the DB2 module that issued the message.

port The TCP/IP port number used by this DB2 subsystem to accept incoming resynchronization connection requests.

System action: The DB2 TCP/IP port is available to accept incoming resynchronization connection requests.

DSNL521I *csect-name* **INVALID DDF BSDS VALUES DETECTED (PORT=***port***,RESPORT=***rport***).**

Explanation: DB2 encountered invalid values for the TCP/IP port numbers in the DDF record of the BSDS. Valid TCP/IP port numbers must obey the following rules:
- If PORT is non-zero, RESPORT must also be non-zero.
- If RESPORT is non-zero, PORT must also be non-zero.
- PORT and RESPORT must not specify the same value.

Variable **Description**

port The TCP/IP port number used by this DB2 subsystem to accept incoming DRDA SQL connection requests.

rport The TCP/IP port number used by this DB2 subsystem to accept incoming 2-phase commit resynchronization requests.

System action: DDF startup terminates abnormally.

System programmer response: Use the Change Log Inventory utility (DSNJU003) to correct the TCP/IP port numbers used by DB2.

DSNL522I *csect-name* **REQUIRED TCP/IP SUPPORT NOT AVAILABLE**

Explanation: The DB2 BSDS indicated that TCP/IP access was required, but the required TCP/IP interfaces were not available on this system. The required TCP/IP interfaces are provided by OS/390 Release 3.

System action: DDF startup terminates abnormally.

System programmer response: Take one of the following actions:

- Install OS/390 Release 3 or later, so that the required TCP/IP interfaces are available.
- Use the Change Log Inventory utility (DSNJU003) to deactivate DB2's TCP/IP support, by specifying PORT=0 and RESPORT=0 on the DDF statement.

DSNL523I *csect* **TCP/IP SERVICES AVAILABLE FOR VIPA** *ipaddr* **AND PORT** *port*

Explanation: DB2 has detected the use of virtual IP address (VIPA) to accept incoming requests. DB2 is available to process connect requests for the specified PORT number bound to the specified IP address.

Variable	Description
csect	The CSECT name of the DB2 module that issued the message.
ipaddr	The specific IP address used by this DB2 subsystem to accept incoming connection requests.
port	The port number used by this DB2 subsystem to accept incoming connection requests.

System action: TCP/IP is initialized and DDF is available to accept incoming requests for the specified IP address. The DDF listener has issued a BIND using INADDR_ANY, but the BIND was intercepted by TCP/IP and converted to BIND specific with the specified IP address. This is triggered by specifying the BIND *ipaddr* on the PORT statement in the PROFILE.TCPIP data set which was used to reserve the PORT for DB2.

If a TCP/IP failure is detected, the DDF listener periodically attempts to re-initialize the TCP/IP listener socket to allow incoming connections. This message is reissued after DDF successfully initializes the listener and is available again to accept incoming requests.

System programmer response: Remove the BIND *ipaddr* parameter on the TCP/IP PORT statement if it is required for DB2 to listen to any IP address supported by the TCP/IP stack.

DSNL600I **RESOURCE UNAVAILABLE CORRELATION ID=***correlation-id* **CONNECTION ID=***connection-id* **LUW ID=***luw-id=token* **REASON=***reason-code* **TYPE=***resource-type* **LOCATION=***location-name* **NAME=***resource-name*

Explanation: This message is produced at the requesting site when a distributed request requires a resource that is not available.

The resource identified by 'resource-name' and 'resource-type' was not available at 'location-name' due to the indicated reason code.

correlation-id
The correlation identifier of the thread associated with the requester of the resource. Correlation IDs beginning with '0nn' where 'nn' is a number from 01 to 28, identify system agents.

connection-id
The connection identifier of the requester.

luw-id=token
The logical unit of work ID of the thread followed by an '=' sign and a token which may be used in place of luw-id in any DB2 command that accepts luw-id as input.

Refer to message DSNT500I under Chapter 17, "DSNT... messages," on page 283 for an explanation of the resource type and resource name.

System action: The SQL statement cannot be executed.

Operator response: See Problem Determination.

Problem determination: The reason the resource is unavailable is identified by the reason code. The reason codes are described in *DB2 Codes*. Collect the information described for the reason code received.

Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1.

DSNL700I *csect-name* **CDB RESOURCE UNAVAILABLE, DEFAULTS MAY BE USED REASON=***reason* **TYPE=***type* **NAME=***name*

Explanation: This message indicates that a requested function required the CDB which is currently unavailable. Defaults can be used to allocate conversations or set conversation limits with VTAM and TCP/IP partners.

System action: If the SYSIBM.MODESELECT table is unavailable, VTAM defaults are used to allocate conversations. If the the SYSIBM.LUMODES table is unavailable, VTAM defaults can be used for conversation limits. If the SYSIBM.LOCATIONS, SYSIBM.IPNAMES, SYSIBM.LULIST, or SYSIBM.LUNAMES tables areunavailable,

conversations can not be allocated to remote LOCATIONs.

Operator response: Note the REASON code, the TYPE code, and the NAME. See Problem Determination section for an explanation of the collected data.

Problem determination: The reason the resource is unavailable is identified by a reason code. The reason codes that appear in this message are described in *DB2 Codes*. Use the table defined for message DSNT500I to find the TYPE of resource identified in the message.

DSNL701I CDB OBJECT *object-name* DOES NOT EXIST

Explanation: The communications database (CDB) does not contain *object-name*. The incorrect *object-name* can be the database name, the table space name, or the table name.

System action: Termination of the distributed data facility (DDF) is initiated.

Operator response: Notify the system programmer.

System programmer response: Determine what caused the failure, define the object as required, and issue the -START DDF command.

Problem determination: Print the definition of the communications database (CDB).

DSNL702I CDB TABLE *table-name* DOES NOT HAVE PROPER COLUMN DEFINITIONS, COLUMN *column-name* IS MISSING OR INCORRECTLY DEFINED

Explanation: The communications database (CDB) needs a definition for 'column-name' in 'tablename'. The current definition is missing or incorrect.

System action: Termination of the distributed data facility (DDF) is initiated.

Operator response: Notify the system programmer.

System programmer response: Determine what caused the failure, define the column as required, and issue the -START DDF command.

Problem determination: Print the definition of the communications database (CDB).

DSNL703I CDB TABLE *table-name* DOES NOT HAVE PROPER INDEX DEFINITIONS, INDEX *index-name* IS MISSING OR INCORRECTLY DEFINED

Explanation: The communications database (CDB) requires an index name for *table-name*. The current definition is missing or incorrect.

System action: The distributed data facility (DDF) terminates.

Operator response: Notify the system programmer.

System programmer response: Define the index as required and issue the -START DDF command. Refer to Part 3 of *DB2 Installation Guide*.

Problem determination: Print the definition of the communications database (CDB).

DSNL704I CDB OBJECT *object-name* DOES NOT HAVE PROPER REFERENTIAL CONSTRAINT DEFINITIONS

Explanation: The communications database (CDB) requires referential constraint definitions for 'object-name'. The current definition is missing or incorrect.

System action: Termination of the distributed data facility (DDF) is initiated.

Operator response: Notify the system programmer.

System programmer response: Determine what caused the failure, define the missing referential constraint, and issue the -START DDF command. Refer to Part 3 of *DB2 Installation Guide* .

Problem determination: Print the definition of the communications database (CDB).

DSNL705I CDB IS UNAVAILABLE OR INCORRECTLY DEFINED

Explanation: During start-up of the distributed data facility, the required CDB was unavailable or incorrectly defined.

System action: Termination of the DDF is initiated.

Operator response: Notify the system programmer.

System programmer response: Refer to the previous message that specifies the cause of the CDB failure (DSNL700I, DSNL701, DSNL702I, DSNL703I, or DSNL704I). Determine what caused the failure, fix the error condition, and issue the -START DDF command.

Collect the following diagnostic items listed in Appendix A, “Problem determination,” on page 659: 1.

DSNL706I *csect-name* AN ERROR HAS OCCURRED IN DDF EXIT ROUTINE *routine*

Explanation: An error occurred following invocation of a DDF exit routine by DB2.

The variable “routine” is the name of the exit routine, DSNLHLIx, invoked by DB2.

System action: The system abends the failing database-access agent.

Chapter 11. DSNM... messages

All of the DSNM... messages are sent to the IMS master terminal operator (MTO), and to the appropriate MVS console.

DSNM001I IMS/TM xxxx CONNECTED TO SUBSYSTEM yyyy

Explanation: This is an information message only. The control region of the IMS/TM system specified in the message as 'xxxx' has successfully connected to the DB2 subsystem specified in the message as 'yyyy'. The message is sent to the IMS/TM master terminal operator.

System action: IMS/TM transactions can access DB2 resources.

DSNM002I IMS/TM xxxx DISCONNECTED FROM SUBSYSTEM yyyy RC=rc

Explanation: The IMS/TM system named 'xxxx' control region has disconnected from the DB2 subsystem named 'yyyy' for the RC, reason code, of 'rc'. The message is sent to the IMS/TM master terminal operator. The reason code values are:

Value	Description
A	IMS/TM is terminating normally (for example, /CKPT FREEZE command).
B	IMS/TM is terminating abnormally.
C	DB2 is terminating in a quiesce fashion (for example, DB2 stop quiesce command).
D	DB2 is terminating abnormally.
E	The connection between IMS/TM and the DB2 subsystem is terminating; however, both subsystems remain operational.

System action: The connection between IMS/TM and DB2 terminated. IMS/TM transactions can no longer access DB2 resources.

Operator response: Notify the system programmer.

System programmer response: Reason code A, C, or E makes this an information message. Reason code D indicates that a serious problem occurred in the DB2 subsystem, and that the SYS1.LOGREC should be examined. For reason code B or D, determine the failing subcomponent, and if you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: SYS1.LOGREC and/or dump information may be available for reason codes B and D.

DSNM003I IMS/TM xxxx FAILED TO CONNECT TO SUBSYSTEM yyyy RC=rc

Explanation: The IMS/TM system (named 'xxxx') control region did not connect to the DB2 subsystem (named 'yyyy'). This message is sent to the IMS/TM master terminal operator. The return code 'rc' indicates the reason for the failure.

Value	Description
00	The DB2 subsystem is not operational. However, a notify message has been queued and, when the DB2 subsystem is started, it will automatically connect to IMS/TM.
04	The connection attempt failed between IMS/TM and the DB2 subsystem.

System action: IMS/TM transactions cannot access DB2 resources.

Operator response: For return code 00, start the DB2 subsystem. For return code 04, notify the system programmer.

System programmer response: For return code 04, refer to Part 4 (Volume 1) of *DB2 Administration Guide* for failure analysis procedures.

Problem determination: For return code 04, an IMS/TM attachment facility snap record (code X'5501FF') is written to the IMS/TM log to indicate the reason for the failure.

DSNM004I RESOLVE INDOUBT ENTRY(S) ARE OUTSTANDING FOR SUBSYSTEM xxxx

Explanation: The DB2 subsystem ID of 'xxxx' has 'indoubt' entry(s) remaining after the IMS/TM subsystem has attempted to resolve the work units that are indoubt. This message is sent to the IMS/TM master terminal operator.

System action: The connection between the DB2 and IMS/TM subsystems remains active, and applications can access DB2 resources.

Operator response: Notify the system programmer.

System programmer response: Refer to Part 4 (Volume 1) of *DB2 Administration Guide* for failure analysis procedures. DB2 commands (DISPLAY THREAD and RECOVER INDOUBT) may be used to display and resolve the work unit(s) that are indoubt.

© Copyright IBM Corp. 1983, 2008

Problem determination: Recovery information is available on the DB2 log and IMS/TM log to assist in the resolution of the work unit(s) indoubt. For each indoubt work unit, an IMS/TM attachment facility snap record (code X'5501FE') is written to the IMS/TM log.

DSNM005I IMS/TM RESOLVE INDOUBT PROTOCOL PROBLEM WITH SUBSYSTEM xxxx

Explanation: The DB2 subsystem ID of 'xxxx' has protocol incompatibilities with IMS/TM. The DB2 resolve indoubt process received an unknown or invalid recovery request (RRE) from IMS/TM. The request was rejected by DB2 and kept by IMS/TM. The message is sent to the IMS/TM master terminal operator.

System action: The connection between the DB2 and IMS/TM subsystems remains active, and applications can access DB2 resources.

Operator response: Notify the system programmer.

System programmer response: Call the support center after extracting from the IMS/TM log the IMS/TM attachment facility snap records (code X'5501FE') using the IMS/TM utility DFSERA10.

These records contain the complete request from IMS/TM to DB2, the request type is at offset X'01C', this is a four-character identifier which should contain any of the following words:

'SHOW', 'IDBT', 'COMM', or 'ABRT'.

If the IMS/TM recovery request (RRE) found in the log is an invalid one, (not 'COMM' or 'ABRT'), you may manually delete it using the IMS/TM/CHANGE SUBSYS RESET command after consulting with the support center.

If the IMS/TM request is a valid one (abort or commit), you must find the action that IMS/TM performed for that RRE by locating the X'3730' (commit) or X'3801' (abort) record in the IMS/TM log and perform the same action under DB2 by using the -RECOVER INDOUBT command. Also refer to Part 4 (Volume 1) of *DB2 Administration Guide* for more information about resolve-indoubt failure analysis.

Problem determination: Recovery information is available on the DB2 log and IMS log to assist in the resolution of the problem. For each indoubt work unit, an IMS/TM attachment facility snap record (code X'5501FE') is written to the IMS/TM log.

DSNM100I SUBSYSTEM xxxx NOT AVAILABLE

Explanation: The IMS/TM command facility which routes commands to the DB2 subsystem cannot perform its functions because the DB2 subsystem named 'xxxx' is considered not operational.

System action: The connection to the DB2 subsystem failed for the command. The command is rejected.

System programmer response: If the DB2 subsystem is operational, and if you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem. The person entering the request may not be authorized to execute the command. If the DB2 subsystem is not operational, DB2 commands from IMS/TM cannot be processed.

Problem determination: A snap record (X'5501FF') is written to the IMS/TM log if a connection request failed.

DSNM104I xxxx COMMAND FAILED

Explanation: The DB2 command specified by 'xxxx' failed to complete.

System action: The command terminated without providing an output message. Commands will still be allowed by the IMS/TM attachment facility.

System programmer response: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: If the command failed because of an abend, a snap record (X'5501FF') is written to the IMS/TM log and SYS1.LOGREC information is available.

Chapter 12. DSNN... message

DSNN40 THE NUMBER YOU SELECTED IS NOT A VALID SQL RETURN CODE NUMBER

Explanation: The user looking at the SQL Return Codes Menu in DB2 HELP selected a number that is not a valid SQL return code.

System action: The SQL Return Codes Menu continues to be displayed.

User response: Select a valid SQL return code number, and try again.

Operator response: Select a valid SQL return code number, and try again.

© Copyright IBM Corp. 1983, 2008

Chapter 13. DSNP... messages

The DSNP-prefixed messages contain a connection ID, a correlation ID, and a logical-unit-of-work ID. Messages that have the same connection ID, correlation ID, and logical-unit-of-work ID relate to the same user. This makes it possible to scan a number of DSNP-prefixed messages and group activities involving a given user.

DSNP001I *csect* – *data-set-name* **IS WITHIN** *nk* **BYTES OF AVAILABLE SPACE. RC=***rrrrrrrr* **CONNECTION-ID=***xxxxxxxx***, CORRELATION-ID=***yyyyyyyyyyyy***, LUW-ID=***logical-unit-of-work-id=token*

Explanation: This message warns of an impending space shortage. The data space manager (DSM) extend function CSECT 'DSNPmmmm' detected the space shortage through the use of a 'hedge' quantity 'n'. Reason code *rrrrrrrr* gives some information about why space could not be obtained. (See *DB2 Codes*.)

There might also be some preceding DSM and VSAM messages that can help to determine why space cannot be obtained. The connection, correlation, and logical-unit-of-work IDs identify the user or application that encountered the problem. Correlation IDs beginning with '0nn', where 'nn' is a number, identify system agents.

A LUW-ID of an asterisk (*) indicates that the thread does not have a LUW-ID. If the LUW-ID is not an asterisk (*), the value is a *logical-unit-of-work-id* that is assigned to the thread followed by an equals (=) sign and a token value, which can be used in place of logical-unit-of-work-id in any DB2 command that accepts logical-unit-of-work-id as input.

Messages that have the same connection ID, correlation ID, and LUW-ID relate to the same user. This makes it possible to scan a number of DSNP-prefixed messages and group activities involving a given user.

System action: If sufficient space is available for the current request, continue. Otherwise, message DSNP007I follows, and the current SQL request fails.

Operator response: DSM messages are directed to the database administrator (DBA) or system administrator. However, the DBA or system administrator can request that the operator issue an MVS WRITELOG operator command (for example, 'W A') to obtain MVS SYSLOG output. The SYSLOG sometimes contains media manager messages that might be helpful to the DBA or system administrator in determining why space could not be allocated.

System programmer response: Based on the reason code (RC) in the message, as well as information that might be available in access method services or VSAM messages preceding this one, try one of the following solutions:

- If you do not anticipate further extension of this table space or index, ignore the message.
- If the table space or index is backed by storage group-defined data sets and this data set is not at maximum extents, use the ALTER STOGROUP statement to add volumes to the storage group (up to 133 of them).
- If the table space or index is backed by 'user-defined' data sets, and not at maximum extents for the data set, issue an ALTER ADDVOLUMES access method services command to provide additional volumes (up to 133 of them). Or, if the table space is nonpartitioned and the high allocated RBA is greater than or equal to 2GB (2,147,483,648) for the indicated data set, define the data set required for the next sequential piece. (Actually, in this case, there should be an indication in console messages or the reason code that the data set required for the next piece was not found.)
- If the data set is at the maximum number of extents, recover or reorganize the table space or index after taking one of the following actions:
  - Enlarge the primary or secondary space allocation quantity of the table space or index.
  - If the DB2 subsystem is running on z/OS V1.7 or later, and the data set is not shared with any z/OS systems at an earlier level, convert the data set to SMS-managed with the Extent Constraint Removal option set to YES in the SMS data class. If you do this, the maximum number of extents is 7257.

  These processes are described in Part 2 (Volume 1) of *DB2 Administration Guide*
- If all of the following conditions are true, the maximum number of extents is 7257:
  - The z/OS level is V1.7 or later
  - The data set is SMS-managed
  - The Extent Constraint Removal option is set to YES in the associated SMS data class
- If any of the conditions outlined in the previous bullet are not true, the maximum number of extents is:
  - 251 to 255 for non-striped data sets
  - 251 to 255 for each stripe for striped data sets

Problem determination: Examine the console messages, SYS1.LOGREC information, and SYSLOG to

© Copyright IBM Corp. 1983, 2008

determine the reason for the condition. Try to resolve it by the methods described in the Operator Response section of this message. If you feel that this message indicates a problem in an IBM program, you should provide the following sources of information about the problem:

- Console sheet showing the message and related messages that might precede or follow it.
- System log printout (obtained via MVS WRITELOG operator command) showing related VSAM/media manager messages that might appear there (usually needed only if you have a message or reason code that indicates a media manager services problem occurred).
- Integrated catalog facility (ICF) catalog listing showing characteristics of the data set in question and of the ICF catalog itself.
- VTOC listing of packs associated with the request, showing space available for data sets, catalogs, VTOC index, and the VTOC itself, as well as data set extent information.
- Any associated dumps.
- SYS1.LOGREC listing showing entries at or near the time of the problem (unless these appear in an associated dump).
- Depending on the reason code, the results of certain SQL SELECT operations on DB2 catalog tables may be needed. (See *DB2 Codes*.)

DSNP002I *csect* **– DEFINE FAILED FOR** *data-set-name* **RC=***rrrrrrrr* **CONNECTION-ID=***xxxxxxxx***, CORRELATION-ID=***yyyyyyyy***, LUW-ID=***logical-unit-of-work-id =token*

Explanation: Data set definition failed (in CSECT 'DSNPmmmm') for the reason identified by RC. (See *DB2 Codes*.) The connection ID, correlation ID, and logical-unit-of-work ID (LUW-ID) identify the user or application that encountered the problem. The connection, correlation, and logical-unit-of-work IDs identify the user or application that encountered the problem. Correlation IDs beginning with '0nn', where 'nn' is a number, identify system agents.

A LUW-ID of '*' indicates that the thread does not have a LUW-ID. If the LUW-ID is not an '*', the value is a logical-unit-of-work-id that is assigned to the thread followed by an '=' sign and a token value, which can be used in place of logical-unit-of-work-id in any DB2 command that accepts logical-unit-of-work-id as input.

Messages that have the same connection ID, correlation ID, and LUW-ID relate to the same user. This makes it possible to scan a number of DSNP-prefixed messages and group activities involving a given user.

System action: The data space manager (DSM) terminates its actions associated with the current SQL request. Higher level functions might terminate the current SQL request or the unit of recovery that issued the request.

Operator response: This message is directed to either the database administrator or the system administrator. Bring the message to their attention if it appears on the MVS operator's console. The installation should consider directing DB2 messages to a console other than the operator's console.

System programmer response: Investigate the problem indicated by the reason code and preceding access method services messages, if any, and correct it, if possible. If an out-of-space condition exists, see the description of message DSNP001I for actions to consider.

Problem determination: See Problem Determination of message DSNP001I for this information.

DSNP003I *csect* **– RESET FAILED FOR** *data-set-name*. **RC=***rrrrrrrr* **CONNECTION-ID=***xxxxxxxx***, CORRELATION-ID=***yyyyyyyyyyyy***, LUW-ID=***logical-unit-of-work-id=token*

Explanation: Reset, a data space manager (DSM) function invoked by DB2 utilities, failed in CSECT 'DSNPmmmm' for the reason identified by RC. (See *DB2 Codes*.) The connection ID, correlation ID, and logical-unit-of-work ID (LUW-ID) identify the user or application that encountered the problem. Correlation IDs beginning with '0nn', where 'nn' is a number, identify system agents.

A LUW-ID of '*' indicates that the thread does not have a LUW-ID. If the LUW-ID is not an '*', the value is a logical-unit-of-work-id that is assigned to the thread followed by an '=' sign and a token value, which can be used in place of logical-unit-of-work-id in any DB2 command that accepts logical-unit-of-work-id as input.

Messages that have the same connection ID, correlation ID, and LUW-ID relate to the same user. This makes it possible to scan a number of DSNP-prefixed messages and tie together activities involving a given user.

System action: DSM terminates its actions associated with the current SQL request. Higher level functions might terminate the current SQL request or the unit of recovery that issued the request.

Operator response: This message is directed to the database administrator or the system administrator. Bring it to that person's attention if it appears on the MVS operator's console. The installation should consider directing DB2 messages to a console other than the operator's console.

System programmer response: Investigate the problem indicated by the reason code and preceding DB2, access method services, or VSAM messages, if any, and correct it, if possible.

It might be necessary to delete and redefine the data set using access method services. Consider making a backup copy before doing this. (Use DB2 utilities to make the copy; in general, VSAM utilities do not work

on DB2 database services data sets.)

Problem determination: See Problem Determination for message DSNP001I for this information.

DSNP004I *csect* **– DELETE FAILED FOR** *data-set-name*. **RC=***rrrrrrrr* **CONNECTION-ID=***xxxxxxxx*, **CORRELATION-ID=***yyyyyyyyyyyy*, **LUW-ID=***logical-unit-of-work-id=token*

Explanation: Deletion failed in CSECT 'DSNPmmmm' for the reason identified by RC. (See *DB2 Codes*.) The connection ID, correlation ID, and logical-unit-of-work ID (LUW-ID) identify the user or application that encountered the problem. Correlation IDs beginning with '0nn', where 'nn' is a number, identify system agents.

The value of the LUWID is alogical-unit-of-work-id that is assigned to the thread followed by an '=' sign and a token value, which can be used in place of logical-unit-of-work-id in any DB2 command that accepts logical-unit-of-work-id as input.

Messages that have the same connection ID, correlation ID, and LUWID relate to the same user. This makes it possible to scan a number of DSNP-prefixed messages and group activities involving a given user.

Severity: 8 (error)

System action: DB2 terminates its actions associated with the current SQL request. DB2 might terminate the current SQL request or the unit of recovery that issued the request.

Operator response: This message is directed to the database administrator or the system administrator. Bring it to that person's attention if it appears on the MVS operator's console. The installation should consider directing DB2 messages to a console other than the operator's console.

System programmer response: Investigate the problem indicated by the reason code and preceding access method services messages if any, and correct it, if possible. Check the data set and delete it by using access method services directly if reasonable to do so. Consider making a backup copy before doing this. (Use DB2 utilities to make the copy; in general, VSAM utilities do not work on DB2 data sets.)

Problem determination: See Problem Determination for message DSNP001I for this information.

DSNP005I *csect* **– COULD NOT REMOVE EXCESS VOLUMES FROM CANDIDATE LIST OF** *data-set-name* **CONNECTION-ID=***xxxxxxxx*, **CORRELATION-ID=***yyyyyyyyyyyy*, **LUW-ID=***logical-unit-of-work-id=token*

Explanation: Volumes from a storage group were entered in the data set's candidate list and could not be removed by CSECT 'DSNPmmmm' after some data space manager (DSM) operation. The connection ID, correlation ID, and logical-unit-of-work ID (LUW-ID) identify the user or application that encountered the problem. Correlation IDs beginning with '0nn', where 'nn' is a number, identify system agents.

A LUW-ID of '*' indicates that the thread does not have a LUW-ID. If the LUW-ID is not an '*', the value is a logical-unit-of-work-id that is assigned to the thread followed by an '=' sign and a token value, which can be used in place of logical-unit-of-work-id in any DB2 command that accepts logical-unit-of-work-id as input.

Messages that have the same connection ID, correlation ID, and LUW-ID relate to the same user. This makes it possible to scan a number of DSNP-prefixed messages and group activities involving a given user.

This is not a major problem. However, the data set will not properly reflect subsequent deletions from the volume list in its storage group (via ALTER STOGROUP command in SQL), because the data set's integrated catalog facility (ICF) catalog entry still contains the volume serial numbers.

If needed, the storage group in question can be identified by issuing an SQL SELECT to get information about the table space or index that is identified by the third and fourth qualifiers (database name, table space name) in the data set name.

System action: Operation continues.

Operator response: This message is directed to the database administrator or the system administrator. Bring it to that person's attention if it appears on the MVS operator's console. The installation should consider directing DB2 messages to a console other than the operator's console.

System programmer response: Look for associated DSM and access method services messages preceding this one to get additional information. Attempt an ALTER REMOVEVOLUMES access method services command, if appropriate, after issuing a LISTCAT to see which volumes are recorded as 'candidate' in the integrated catalog facility (ICF) catalog.

Problem determination: See Problem Determination for message DSNP001I for this information. (However, there should not be a need for VTOC or SYSLOG information for this case.)

DSNP006I *csect* **– REQUIRED STOGROUP DEFINED DATA SET NOT AVAILABLE—** *data-set-name* **RC=***rrrrrrrr* **CONNECTION-ID=***xxxxxxxx*, **CORRELATION-ID=***yyyyyyyyyyyy*, **LUW-ID=***logical-unit-of-work-id=token*

Explanation: An attempt to alter the shareoptions of this data set failed. The data set could not be accessed for the reason indicated by RC. (See *DB2 Codes*.) The connection ID, correlation ID, and logical-unit-of-work

ID (LUWID) identify the user or application that encountered the problem. Correlation IDs beginning with '0nn', where 'nn' is a number, identify system agents.

The value of the LUWIDis a logical-unit-of-work-id that is assigned to the thread followed by an '=' sign and a token value, which can be used in place of the logical-unit-of-work-id in any DB2 command that accepts the logical-unit-of-work-id as input.

Messages having the same connection ID, correlation ID, and LUWID relate to the same user. This makes it possible to scan a number of DSNP-prefixed messages and group activities involving a given user.

Severity: 8 (error)

System action: DB2 terminates its actions associated with the current SQL request. DB2 might terminate the current SQL request or the unit of recovery that issued the request.

Operator response: This message is directed to the database administrator or the system administrator. Bring it to that person's attention if it appears on the MVS operator's console. The installation should consider directing DB2 messages to a console other than the operator's console.

System programmer response: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: See Problem Determination for message DSNP001I for this information.

DSNP007I *csect* **– EXTEND FAILED FOR** *data-set-name*. **RC=***rrrrrrrr* **CONNECTION-ID=***xxxxxxxx*, **CORRELATION-ID=***yyyyyyyyyyyy*, **LUW-ID=***logical-unit-of-work-id =token*

Explanation: Extension failed in CSECT 'DSNPmmmm' of the data space manager (DSM). The 'data-set-name' identifies the data set that is being extended. RC is the reason code, which indicates why the extend process failed. (See *DB2 Codes*.) The 'connection ID', 'correlation ID', and 'logical-unit-of-work-id' (LUW-ID) identify the user or application that encountered the problem. Correlation IDs beginning with '0nn', where 'nn' is a number, identify system agents.

A LUW-ID of '*' (asterisk) indicates that the thread does not have a LUW-ID. If the 'LUW-ID' is not an asterisk ('*'), the value is a 'logical-unit-of-work-id' that is assigned to the thread followed by an '=' (equal sign) and a token value, which can be used in place of 'logical-unit-of-work-id' in any DB2 command that accepts 'logical-unit-of-work-id' as input.

Messages that have the same 'connection ID', 'correlation ID', and 'LUW-ID' relate to the same user. This situation makes it possible to scan a number of DSNP-prefixed messages and group activities involving a given user.

System action: The DSM terminates its actions associated with this extend request. Higher level functions might terminate if they cannot continue without a successful extend. In this case, additional console messages and/or SQL codes are produced. In other cases, higher level functions might be able to resolve the space need without the extend, so they continue normally.

System programmer response: The following sources of information are available to help determine the cause of the failure:

- See the Programmer Response for message DSNP001I.
- Analyze the RC value in conjunction with the console messages.
- The reason code might apply to the data set identified in the message or to another data set that is needed for the extend process. If the reason code indicates a data set access failure, the console messages indicate the data set name. For example, if the RC indicates 'resource unavailable', console messages identify the resource that is unavailable. Most likely, the message will be DSNT500I.
- Retrieve diagnostic information from the output of the DISPLAY DB...RESTRICT or DISPLAY UTIL(*)... commands.
- If the extend failed on a result table space, refer to "Sort Data Sets" in Part 5 (Volume 2) of *DB2 Administration Guide*.
- If the extend failed on an SQL insert or update, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.
- If you need to enlarge your data set, refer to "Out of disk space or extent limit recovery" in Part 4 (Volume 1) of *DB2 Administration Guide*.
- To find out how the extend function operates, refer to "Improving disk utilization: space and device utilization" in Part 5 (Volume 2) of *DB2 Administration Guide*.

Operator response: See Operator Response for message DSNP001I for this information.

Problem determination: See Problem Determination for message DSNP001I.

DSNP008I *csect*– **REQUIRED USER–DEFINED DATA SET NOT AVAILABLE—***data-set-name* **RC=***rrrrrrrr* **CONNECTION-ID=***xxxxxxxx*, **CORRELATION-ID=***yyyyyyyyyyyy*, **LUW-ID=***logical-unit-of-work-id=token*

Explanation: One of the following actions applied against user-defined data sets failed in CSECT

'DSNPmmmm' of the data space manager (DSM):

- Create table space or index
- Extend table space or index
- Process the SQL ALTER DATABASE statement
- Process the REORG TABLESPACE utility with SHRLEVEL REFERENCE or CHANGE

Correlation IDs beginning with '0nn', where 'nn' is a number, identify system agents.

One or more of the data sets could not be accessed for the reason indicated by RC. (See *DB2 Codes*.) The connection ID, correlation ID, and logical-unit-of-work ID (LUWID) identify the user or application that encountered the problem.

The value of the LUWID is alogical-unit-of-work-id that is assigned to the thread followed by an '=' sign and a token value, which can be used in place of logical-unit-of-work-id in any DB2 command that accepts logical-unit-of-work-id as input.

Messages having the same connection ID, correlation ID, and LUWID relate to the same user. This makes it possible to scan a number of DSNP-prefixed messages and group activities involving a given user.

If the name of the data set in the message is different from the name of the original data set (for example, '....A002' instead of '...A001'), a new data set might need to be created. A different name indicates that the original data set exceeded its limit.

Severity: 8 (error)

System action: DB2 terminates its actions associated with the current SQL request or utility. DB2 might terminate the current SQL request or utility or the unit of recovery that issued the request.

Operator response: This message is directed to the database administrator or the system administrator. Bring it to that person's attention if it appears on the MVS operator's console. The installation should consider directing DB2 messages to a console other than the operator's console.

System programmer response: Refer to the reason code in *DB2 Codes*. The return code description might indicate that one of the following actions needs to be performed or it indicates the necessary steps:

- If the data set does not exist, create it before attempting the create table space/index or extend data set again.
- If the data set shareoptions are incorrect, modify the shareoptions with access method services (AMS).
- The appropriate AMS command is ALTER SHAREOPTIONS.

Problem determination: See Problem Determination for message DSNP001I for this information.

DSNP009I THE FOLLOWING ERROR MESSAGES WERE RECEIVED FOR *operation* **ON** *data-set-name*

DSNP010I END OF ERROR MESSAGES. CONNECTION-ID=*xxxxxxxx***, CORRELATION-ID=***yyyyyyyyyyyy***, LUW-ID=***logical-unit-of-work-id=token***

Explanation: Messages DSNP009I and DSNP010I enclose a block of messages that are received by the data space manager (DSM) subcomponent of DB2 in response to an access method services (AMS) operation.

operation
The access method services operation that was being performed.

data-set-name
The data set on which the operation was being performed.

The prefix of the messages in the block is usually IDC (access method services messages). However, the prefix can also be IKJ (dynamic allocation) or other system subcomponent IDs. Refer to the appropriate MVS publication for descriptions of the enclosed messages.

The operation can be DEFINE CLUSTER, ALTER
ADDVOLUMES, ALTER REMOVEVOLUMES, ALTER
NEWNAME, or DELETE CLUSTER. One or more other
DSM messages follow this message to explain what
happened as a result of the failure of this access
method services request. The connection ID, correlation
ID, and logical-unit-of-work ID (LUW-ID) identify the
user or application that encountered the problem.

A LUW-ID of '*' indicates that the thread does not have a LUW-ID. If the LUW-ID is not an '*', the value is a logical-unit-of-work-id that is assigned to the thread followed by an '=' sign and a token value, which can be used in place of logical-unit-of-work-id in any DB2 command that accepts logical-unit-of-work-id as input.

Messages that have the same connection ID, correlation ID, and LUW-ID relate to the same user. This makes it possible to scan a number of DSNP-prefixed messages and group activities involving a given user. Correlation IDs beginning with '0nn', where 'nn' is a number, identify system agents.

System action: Operation continues.

Operator response: This message is directed to the database administrator or the system administrator. Bring it to that person's attention if it appears on the MVS operator's console. The installation should consider directing DB2 messages to a console other than the operator's console.

System programmer response: Use this information to help resolve problems indicated by other DSM messages that follow this one.

Problem determination: See Problem Determination for message DSNP001I for this information.

DSNP011I *csect* **– MEDIA MANAGER SERVICES ERROR FOR** *data-set-name*. **MMRC=***cccccccc* **DSMRC=***ffffffff* **CONNECTION-ID=***xxxxxxxx*, **CORRELATION-ID=***yyyyyyyyyyyy*, **LUW-ID=***logical-unit-of-work-id =token*

Explanation: The MMRC is the return code that DB2 received from media manager services. Refer to the appropriate MVS publication for a description of these codes. DSMRC is a DSM reason code that indicates which of the media manager services functions, update or extend, failed. (See *DB2 Codes*.) The message is from CSECT 'DSNPmmmm' of the data space manager (DSM). The connection ID, correlation ID, and logical-unit-of-work ID (LUW-ID) identify the user or application that encountered the problem. Correlation IDs beginning with '0nn', where 'nn' is a number, identify system agents.

A LUW-ID of '*' indicates that the thread does not have a LUW-ID. If the LUW-ID is not an '*', the value is a logical-unit-of-work-id that is assigned to the thread followed by an '=' sign and a token value, which can be used in place of logical-unit-of-work-id in any DB2 command that accepts logical-unit-of-work-id as input.

Messages that have the same connection ID, correlation ID, and LUW-ID relate to the same user. This makes it possible to scan a number of DSNP-prefixed messages and group activities involving a given user.

System action: The DSM terminates its actions associated with the current SQL request. Higher level functions might terminate the current SQL request or the unit of recovery that issued the request.

Operator response: DSM messages are directed to either the database administrator (DBA) or the system administrator. However, they can request that the operator issue an MVS WRITELOG operator command (for example, 'W A') to obtain MVS SYSLOG output. The SYSLOG sometimes contains media manager messages that might be helpful to the DBA or system administrator in determining why space could not be allocated.

System programmer response: This message might be preceded by one or more IEC or ICY messages on the console or in SYSLOG. These messages help to determine what happened and what action to take.

Problem determination: See Problem Determination for message DSNP001I for this information.

DSNP012I *csect* **– ERROR IN ICF CATALOG LOCATE FUNCTION FOR** *data-set-name* **CTLGRC=***cc* **CTLGRSN=***zzzzrrss* **CONNECTION-ID=***xxxxxxxx*, **CORRELATION-ID=***yyyyyyyyyyyy*, **LUW-ID=***logical-unit-of-work-id=token*

Explanation: CTLGRC and CTLGRSN are the return and reason codes that DB2 received from SVC 26 call (ICF catalog locate function). The reason code is in the same format as that returned to DB2 by the SVC 26 call.

If CLTGRC is 4 or 8, CTLGRSN is associated with access method services message IDC3009I. To determine the name of the CSECT that set CTLGRSN, convert *zzzz* from hexadecimal to EBCDIC, and append the resulting two characters to 'IGG0CL'. For example, suppose that *zzzz* is C6C6. The EBCDIC equivalent of X'C6C6' is 'FF', so the CSECT that set the reason code is IGG0CLFF.

rr is the hexadecimal equivalent of the reason code value that is documented in access method services message IDC3009I. *ss* is the hexadecimal equivalent of the return code value that is documented in access method services message IDC3009I (the return code follows the reason code.) Convert *rr* and *ss* to decimal. Then see the description of IDC3009I for explanations of the reason code and return code. For example, suppose that *rrss* is 0532. The decimal equivalent of X'32' is 50, and the decimal equivalent of X'05' is 5. Therefore, in IDC009I, you need to look up a return code of 50 and a reason code of 5.

If CTLGRC=AAAAAA08 and CTLGRSN=AAAAAA08, CTLGRSN is not associated with access method services message IDC3009I. AAAAAA08 indicates that data-set-name is not in the ICF catalog.

The connection ID, correlation ID, and LUW-ID identify the user or application that encountered the problem. Correlation IDs beginning with '0nn', where 'nn' is a number, identify system agents.

An LUW-ID with an asterisk (*) indicates that the thread does not have an LUW-ID. If the LUW-ID is not an asterisk (*), the value is a *logical-unit-of-work-id* that is assigned to the thread followed by an equals sign (=) and a token value, which can be used in place of *logical-unit-of-work-id* in any DB2 command that accepts *logical-unit-of-work-id* as input.

Messages that have the same connection ID, correlation ID, and LUW-ID relate to the same user. This makes it possible to scan a number of DSNP-prefixed messages and group activities involving a given user.

System action: DB2 terminates its actions associated with the current SQL request. Higher level functions might terminate the current SQL request or the unit of recovery that issued the request.

System programmer response: Correct the problem indicated by the catalog return and reason codes.

Problem determination: See Problem Determination for message DSNP001I for this information.

DSNP013I **INSUFFICIENT PRIMARY ALLOCATION SPACE TO FORMAT CONTROL PAGES IN DATA SET** *data-set-name* **CONNECTION-ID=***xxxxxxxx***, CORRELATION-ID=***yyyyyyyyyyyy***, LUW-ID=***logical-unit-of-work-id =token*

Explanation: Data space manager (DSM) discovered that the primary space quantity given in the data set definition was not large enough. Formatting of required control information in the first several pages was not possible. The connection ID, correlation ID, and logical-unit-of-work ID (LUW-ID) identify the user or application that encountered the problem.

A LUW-ID of '*' indicates that the thread does not have a LUW-ID. If the LUW-ID is not an '*', the value is a logical-unit-of-work-id that is assigned to the thread followed by an '=' sign and a token value, which can be used in place of logical-unit-of-work-id in any DB2 command that accepts logical-unit-of-work-id as input.

Messages that have the same connection ID, correlation ID, and LUW-ID relate to the same user. This makes it possible to scan a number of DSNP-prefixed messages and group activities involving a given user. Correlation IDs beginning with '0nn', where 'nn' is a number, identify system agents.

System action: DSM terminates its actions associated with the current SQL request. Higher level functions might terminate the current SQL request or the unit of recovery that issued the request.

Operator response: This message is directed to the database administrator or the system administrator. Bring it to that person's attention if it appears on the MVS operator's console. The installation should consider directing DB2 messages to a console other than the operator's console.

System programmer response: This should occur only for table spaces or indexes that are not storage group defined. (The DB2 interpreter forces the initial allocation to be large enough if a storage group is used.) Check both the page size specification (4KB or 32KB) and the number of control pages required (2 for table space, 3 for index space). Specify the primary allocation value again so that at least (pages × page size) bytes are allocated.

Problem determination: See Problem Determination for message DSNP001I for this information.

DSNP014I **IRLM MANUAL LOCK REQUEST FAILED FOR** *data-set-name***. RC=***rr***,SC=***ssssssss***, CONNECTION-ID=***xxxxxxxx***, CORRELATION-ID=***yyyyyyyyyyyy***, LUW-ID=***logical-unit-of-work-id=token***

Explanation: DB2 requested a manual lock on a data set, but the lock could not be obtained for reasons specified in RC and SC. RC and SC are the return code and reason code from IRLM. RC and the first two digits of SC are documented in *DB2 Codes*. The connection ID, correlation ID, and logical-unit-of-work ID (LUW-ID) identify the user or application that encountered the problem.

A LUW-ID of '*' indicates that the thread does not have a LUW-ID. If the LUW-ID is not an '*', the value is a logical-unit-of-work-id that is assigned to the thread followed by an '=' sign and a token value, which can be used in place of logical-unit-of-work-id in any DB2 command that accepts logical-unit-of-work-id as input.

Messages that have the same connection ID, correlation ID, and LUW-ID relate to the same user. This makes it possible to scan a number of DSNP-prefixed messages and group activities involving a given user. DSNP001I or DSNP007I follow this message. Correlation IDs beginning with '0nn', where 'nn' is a number, identify system agents.

If message DSNT376I was received before message DSNP014I, the lock request failed because of a timeout condition. Otherwise, refer to the IRLM return and reason codes for an explanation of the failure.

System action: The current SQL or the unit of recovery that issued the request might be terminated.

Operator response: This message is directed to the database administrator or the system administrator. Bring it to that person's attention if it appears on the MVS operator's console. The installation should consider directing DB2 messages to a console other than the operator's console.

System programmer response: Investigate the problem indicated by the IRLM return and reason code.

Problem determination: Provide the console sheet and a dump. A GTF trace might also be required. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem. Refer to the explanation section of this message for additional problem determination information.

DSNP015I *csect* **– IRLM MANUAL UNLOCK REQUEST FAILED FOR** *data-set-name***. RC=***rr***,SC=***ssssssss***, CONNECTION-ID=***xxxxxxxx***, CORRELATION-ID=***yyyyyyyyyyyy***, LUW-ID=***logical-unit-of-work-id=token***

Explanation: DB2 requested a manual unlock on a data set, but the unlock could not be performed for reasons specified in RC and SC. RC and SC are the return code and reason code from IRLM. RC and the first two digits of SC are documented in *DB2 Codes*. The connection ID, correlation ID, and logical-unit-of-work ID (LUW-ID) identify the user or application that encountered the problem.

A LUW-ID of '*' indicates that the thread does not have

a LUW-ID. If the LUW-ID is not an '*', the value is a logical-unit-of-work-id that is assigned to the thread followed by an '=' sign and a token value, which can be used in place of logical-unit-of-work-id in any DB2 command that accepts logical-unit-of-work-id as input.

Messages that have the same connection ID, correlation ID, and LUW-ID relate to the same user. This makes it possible to scan a number of DSNP-prefixed messages and group activities involving a given user. Correlation IDs beginning with '0nn', where 'nn' is a number, identify system agents.

System action: The current SQL or the unit of recovery that issued the request might be terminated.

Operator response: This message is directed to the database administrator or the system administrator. Bring it to that person's attention if it appears on the MVS operator's console. The installation should consider directing DB2 messages to a console other than the operator's console.

System programmer response: Investigate the problem indicated by the IRLM return and reason code.

Problem determination: Provide the console sheet and a dump. A GTF trace might also be required. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem. See the explanation section of this message for additional problem determination information.

DSNP016I **CREATE FAILED FOR** *pageset-name*. **RC=***rrrrrrrr* **CONNECTION-ID=***xxxxxxxx*, **CORRELATION-ID=***yyyyyyyyyyyy*, **LUW-ID=***logical-unit-of-work-id =token*

Explanation: Creation of a table space or index space failed in the data space manager (DSM) for the reason given in RC. (See *DB2 Codes*.) The pageset-name format is 'dbname.tiname.instance', where 'dbname' is the database name and 'tiname' is the table space or index space name.

This message identifies the function that was in progress at the time of a failure. The specific failure is usually identified by a preceding message. The connection ID, correlation ID, and logical-unit-of-work ID (LUW-ID) identify the user or application that encountered the problem.

A LUW-ID of '*' indicates that the thread does not have a LUW-ID. If the LUW-ID is not an '*', the value is a logical-unit-of-work-id that is assigned to the thread followed by an '=' sign and a token value, which can be used in place of logical-unit-of-work-id in any DB2 command that accepts logical-unit-of-work-id as input.

Messages that have the same connection ID, correlation ID, and LUW-ID relate to the same user. This makes it possible to scan a number of DSNP-prefixed messages and group activities involving a given user. Correlation IDs beginning with '0nn', where 'nn' is a number, identify system agents.

System action: DSM terminates its actions associated with the current SQL request. Higher level functions might terminate the current SQL request or the unit of recovery that issued the request.

Operator response: This message is directed to the database administrator or the system administrator. Bring it to that person's attention if it appears on the MVS operator's console. The installation should consider directing DB2 messages to a console other than the operator's console.

System programmer response: Investigate the problem indicated by reason code and preceding messages, and correct it, if possible.

Problem determination: See Problem Determination for message DSNP001I for this information.

DSNP017I **RESET FAILED FOR** *pageset-name*. **RC=***rrrrrrrr* **CONNECTION-ID=***xxxxxxxx*, **CORRELATION-ID=***yyyyyyyyyyyy*, **LUW-ID=***logical-unit-of-work-id =token*

Explanation: Reset failed in the data space manager (DSM) subcomponent of DB2 for a table space or index space for the reason given in RC. (See *DB2 Codes*.) The 'pageset-name' format is 'dbname.tiname', where 'dbname' is the database name and 'tiname' is the table space or index space name.

This message identifies the function that was in progress at the time of a failure. The specific failure is usually identified by a preceding message. The connection ID, correlation ID, and logical-unit-of-work ID (LUW-ID) identify the user or application that encountered the problem.

A LUW-ID of '*' indicates that the thread does not have a LUW-ID. If the LUW-ID is not an '*', the value is a logical-unit-of-work-id that is assigned to the thread followed by an '=' sign and a token value, which can be used in place of logical-unit-of-work-id in any DB2 command that accepts logical-unit-of-work-id as input.

Messages that have the same connection ID, correlation ID, and LUW-ID relate to the same user. This makes it possible to scan a number of DSNP-prefixed messages and group activities involving a given user. Correlation IDs beginning with '0nn', where 'nn' is a number, identify system agents.

System action: DSM terminates its actions associated with the current SQL request. Higher level functions might terminate the current SQL request or the unit of recovery that issued the request.

Operator response: This message is directed to the database administrator or the system administrator. Bring it to that person's attention if it appears on the MVS operator's console. The installation should

consider directing DB2 messages to a console other than the operator's console.

System programmer response: Investigate the problem indicated by the reason code and preceding messages, and correct it, if possible.

Problem determination: See Problem Determination for message DSNP001I for this information.

DSNP018I **DROP FAILED FOR** *pageset-name*. **RC=***rrrrrrrr* **CONNECTION-ID=***xxxxxxxx*, **CORRELATION-ID=***yyyyyyyyyyyy*, **LUW-ID=***logical-unit-of-work-id =token*

Explanation: Drop failed in the data space manager (DSM) subcomponent of DB2 for a table space or index space for the reason given in RC. Refer to *DB2 Codes*. The 'pageset-name' format is 'dbname.tiname', where 'dbname' is the database name and 'tiname' is the table space or index space name. This message identifies the function that was in progress at the time of a failure. The specific failure is usually identified by a preceding message. The connection ID, correlation ID, and logical-unit-of-work ID (LUW-ID) identify the user or application that encountered the problem.

A LUW-ID of '*' indicates that the thread does not have a LUW-ID. If the LUW-ID is not an '*', the value is a logical-unit-of-work-id that is assigned to the thread followed by an '=' sign and a token value, which can be used in place of logical-unit-of-work-id in any DB2 command that accepts logical-unit-of-work-id as input.

Messages that have the same connection ID, correlation ID, and LUW-ID relate to the same user. This makes it possible to scan a number of DSNP-prefixed messages and group activities involving a given user. Correlation IDs beginning with '0nn', where 'nn' is a number, identify system agents.

System action: DSM terminates its actions associated with the current SQL request. Higher level functions might terminate the current SQL request or the unit of recovery that issued the request.

Operator response: This message is directed to the database administrator or the system administrator. Bring it to that person's attention if it appears on the MVS operator's console. The installation should consider directing DB2 messages to a console other than the operator's console.

System programmer response: Investigate the problem indicated by the reason code and preceding messages, and correct it, if possible.

Problem determination: See Problem Determination for message DSNP001I for this information.

DSNP019I *csect* – **REALLOCATE FAILED FOR** *data-set-name*. **RC=***rrrrrrrr* **CONNECTION-ID=***xxxxxxxx*, **CORRELATION-ID=***yyyyyyyyyyyy*, **LUW-ID=***logical-unit-of-work-id =token*

Explanation: After a successful ALTER REMOVEVOLUMES request, the data space manager (DSM) subcomponent of DB2 was unable to reallocate the data set for the reason indicated by RC. (See *DB2 Codes*.) The connection ID, correlation ID, and logical-unit-of-work ID (LUW-ID) identify the user or application that encountered the problem.

A LUW-ID of '*' indicates that the thread does not have a LUW-ID. If the LUW-ID is not an '*', the value is a logical-unit-of-work-id that is assigned to the thread followed by an '=' sign and a token value, which can be used in place of logical-unit-of-work-id in any DB2 command that accepts logical-unit-of-work-id as input.

Messages that have the same connection ID, correlation ID, and LUW-ID relate to the same user. This makes it possible to scan a number of DSNP-prefixed messages and group activities involving a given user. Correlation IDs beginning with '0nn', where 'nn' is a number, identify system agents.

System action: Operation continues.

Operator response: This message is directed to the database administrator or the system administrator. Bring it to that person's attention if it appears on the MVS operator's console.

System programmer response: The data set and corresponding page set piece or partition can usually continue to be used by the DB2 subsystem. However, they should be stopped and restarted at the earliest possible time in order to synchronize the allocation control blocks with the integrated catalog facility (ICF) catalog volume list.

Problem determination: See Problem Determination for message DSNP001I for this information.

DSNP020I *csect-name* **- USER DEFINED ATTRIBUTES INCORRECT FOR** *data-set-name*. **RC=***rrrrrrrr* **CONNECTION-ID=***xxxxxxxx*, **CORRELATION-ID=***yyyyyyyy*, **LUW-ID=***logical-unit-of-work-id =token*

Explanation: The specified data set was access method services user-defined and does not have the required DB2 attributes. The connection ID, correlation ID, and LUW-ID identify the user or application that encountered the problem.

The value of the LUWID is a'logical-unit-of-work-id' that is assigned to the thread, followed by an '=' sign and a token value, which can be used in place of 'logical-unit-of-work-id' in any DB2 command that accepts 'logical-unit-of-work-id' as input.

Messages having the same connection ID, correlation ID, and LUWID relate to the same user. This makes it possible to scan a number of DSNP-prefixed messages and group activities involving a given user. Correlation IDs beginning with '0nn', where 'nn' is a number, identify system agents.

Severity: 8 (error)

System action: No action is taken with the named data set. An error code is returned and a dump is not taken.

Operator response: Notify the system programmer.

System programmer response: Delete the named data set and define it with the correct attributes.

Problem determination: Obtain an access method services LISTCAT for the named data set's attributes. Refer to Part 2 (Volume 1) of *DB2 Administration Guide* for the correct data set definition attributes.

DSNP021I *csect-name* **EXTEND NOTIFY FAILED FOR** *data-set-name*, **SC=***ssssssss*, **NOTESC=***nnnnnnnn*, **CONNECTION-ID=***xxxxxxxx*, **CORRELATION-ID=***yyyyyyyyyyyy*, **LUW-ID=***logical-unit-of-work-id=token*

Explanation: This is an internal error. The connection ID, correlation ID, and logical-unit-of-work ID (LUW-ID) identify the user or application that encountered the problem.

An LUW-ID of '*' indicates that the thread does not have an LUW-ID. If the LUW-ID is not an '*', the value is a logical-unit-of-work-id that is assigned to the thread followed by an '=' sign and a token value, which can be used in place of the logical-unit-of-work-id in any DB2 command that accepts the logical-unit-of-work-id as input.

Messages that have the same connection ID, correlation ID, and LUW-ID relate to the same user. This makes it possible to scan a number of DSNP-prefixed messages and group activities involving a given user. Correlation IDs beginning with '0nn', where 'nn' is a number, identify system agents.

If message DSNT376I was received before message DSNP021I, the notify request failed because of a timeout condition.

System action: The current SQL statement or the unit of recovery that issued the request might be terminated.

Problem determination: Obtain the console sheet and a dump. A GTF trace might also be required.

Refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNP022I *csect-name* **EXTEND DBET RECORDING FAILED FOR** *data-set-name*, **RC=***rrrrrrrr*, **SC=***ssssssss*, **DBETSC=***nnnnnnnn*, **CONNECTION-ID=***xxxxxxxx*, **CORRELATION-ID=***yyyyyyyyyyyy*, **LUW-ID=***logical-unit-of-work-id=token*

Explanation: This is an internal error. The connection ID, correlation ID, and logical-unit-of-work ID (LUW-ID) identify the user or application that encountered the problem.

An LUW-ID of '*' indicates that the thread does not have an LUW-ID. If the LUW-ID is not an '*', the value is a logical-unit-of-work-id that is assigned to the thread followed by an '=' sign and a token value, which can be used in place of the logical-unit-of-work-id in any DB2 command that accepts the logical-unit-of-work-id as input.

Messages that have the same connection ID, correlation ID, and LUW-ID relate to the same user. This makes it possible to scan a number of DSNP-prefixed messages and group activities involving a given user. Correlation IDs beginning with '0nn', where 'nn' is a number, identify system agents.

System action: The current SQL statement or the unit of recovery that issued the request might be terminated.

Problem determination: Obtain the console sheet and a dump. A GTF trace might also be required.

Refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNP026I *csect-name* **DATASET IS NOT ALLOWED TO BE DELETED** *data-set-name*, **RC=***rrrrrrrr*, **CONNECTION-ID=***xxxxxxxx*, **CORRELATION-ID=***yyyyyyyyyyyy*, **LUW-ID=***logical-unit-of-work-id=token*

Explanation: During clean up of the shadow data set in the final step of online REORG, an error is detected due to an attempt to delete a regular table or index space data set. Refer to Part 2 of *DB2 SQL Reference* for information about online REORG. The connection ID, correlation ID, and logical unit of work ID (LUWID) identify the user or application that encountered the problem. Correlation IDs beginning with '0nn', where 'nn' is a number, identify system agents.

The value of the LUWID is assigned to the thread followed by an '=' sign and a token value, that can be used in place of the LUWID in any DB2 command that accepts it as input.

Messages that have the same connection ID, correlation ID, and LUWID relate to the same user. This makes it possible to scan a number of DSNP-prefixed messages and group activities involving a given user.

System action: No action is taken with the named

data set. An error code is returned and a dump is not taken.

Operator response: Notify the system programmer.

System programmer response: Determine the failing environment. Refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: This is an internal error in the online REORG of utility. See the 'Problem Determination' section of message DSNP001I for this information.

DSNP027I *csect-name* **RENAME FAILED OLD DATASET=***data-set-name***, NEW DATASET=***data-set-name***, RC=***rrrrrrrr***, CONNECTION-ID=***xxxxxxxx***, CORRELATION-ID=***yyyyyyyyyyyy***, LUW-ID=***logical-unit-of-work-id=token***

Explanation: An attempt to alter the data set name failed. The connection ID, correlation ID, and logical unit of work ID (LUWID) identify the user or application that encountered the problem. Correlation IDs beginning with '0nn', where 'nn' is a number, identify system agents.

The value of the LUWID is assigned to the thread followed by an '=' sign and a token value, that can be used in place of the LUWID in any DB2 command that accepts it as input.

Messages that have the same connection ID, correlation ID, and LUWID relate to the same user. This makes it possible to scan a number of DSNP-prefixed messages and group activities involving a given user.

Severity: 8 (error)

System action: DB2 terminates its actions associated with the current SQL request. DB2 might terminate the current SQL request or the unit of recovery that issued the request.

Operator response: This message is directed to the database administrator or the system administrator. Bring it to that person's attention if it appears on the MVS operator's console. The installation should consider directing DB2 messages to a console other than the operator's console.

System programmer response: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: See 'Problem Determination' for message DSNP001I for this information.

DSNP028I *csect-name* **HSM RECALL FAILED FOR DSNAME =** *dsn* **HSMRC=***rrrrrrrr***,HSMRSN=***ssssssss* **CONNECTION ID=***xxxxxxxx***, CORRELATION ID=***yyyyyyyyyyyy***, LUW-ID=***logical-unit-of-work-id =token***

Explanation: This message indicates that a required data set was migrated by HSM and an attempt to recall the data set failed, preventing DB2 from accessing the data set. The HSMRC and HSMRSN fields can be used to determine the HSM failure. See Problem Determination below. The connection ID, correlation ID, and 'logical-unit-of-work ID' (LUW-ID) identify the user or application that encountered the problem.

A LUW-ID of '*' indicates that the thread does not have a LUW-ID. If the LUW-ID is not an '*', the value is a 'logical-unit-of-work-id' that is assigned to the thread followed by an '=' sign and a token value, which can be used in place of 'logical-unit-of-work-id' in any DB2 command that accepts 'logical-unit-of-work-id' as input.

Messages that have the same connection ID, correlation ID, and LUW-ID relate to the same user. This makes it possible to scan a number of DSNP-prefixed messages and group activities involving a given user. Correlation IDs beginning with '0nn', where 'nn' is a number, identify system agents.

System action: DSM terminates its action with the associated data set. Higher level functions that issued the recall request might terminate.

Operator response: Notify the system programmer.

System programmer response: Investigate the reason for the HSM recall failure. Use the information in this message to help resolve this problem. Additional messages on the MVS operator's console might be associated with this message.

Problem determination: If the HSMRSN is 999, refer to the following list for the explanation:

HSMRC

	Description
4	HSM is not active.
8	No storage is available for MWE in CSA.
36	Input MWE is not within user storage.
40	Allocated amount of CSA for MWEs was exceeded.
100	There was an invalid function used.

For all other reason codes (HSMRSN), refer to the *z/OS DFSMShsm Managing Your Own Data* for an explanation of HSM user messages. The HSM user messages can be determined by appending the value in the HSMRC field of this message to ARC1100I (that is, HSMRC=12, HSM message is ARC1112I). The HSMRSN reason code is referenced in the ARC1100I messages.

DSNP030I *csect-name* **THE CURRENT EXTENT REACHES THE LAST VOLUME FOR** *data-set-name***. CONNECTION-ID=***xxxxxxxx***, CORRELATION-ID=***yyyyyyyyyyyy***, LUW-ID=***logical-unit-of-work-id***=***token***

Explanation: This message indicates that the last volume to which the data set *data-set-name* can extend has been reached. You cannot add more volumes if the current volume runs out of space.

The connection, correlation, and logical-unit-of-work IDs identify the user or application that encountered the problem. Correlation IDs beginning with '0nn', where 'nn' is a number, identify system agents.

A LUW-ID of '*' indicates that the thread does not have a LUW-ID. If the LUW-ID is not an '*', the value is a *logical-unit-of-work-id* that is assigned to the thread followed by an '=' sign and a *token* value, which can be used in place of *logical-unit-of-work-id* in any DB2 command that accepts logical-unit-of-work-id as input.

Messages that have the same connection ID, correlation ID, and LUW-ID relate to the same user. This makes it possible to scan a number of DSNP-prefixed messages and group activities involving a given user.

Severity: 0 (informational)

System action: If sufficient space in current volume is available for the current request, DB2 continues. Otherwise, message DSNP007I is issued, and the current SQL request fails.

Operator response: For more information, issue an MVS WRITELOG operator command (for example, 'W A') to obtain MVS SYSLOG output. The SYSLOG sometimes contains VSAM or media manager messages that might be helpful in determining why space could not be allocated.

System programmer response: Depending on what information is available in the access method services or VSAM messages preceding this one, try one of the following:

- If you do not anticipate further extension of this table space or index, ignore the message.
- If the table space or index is backed by SMS-managed data sets, use LISTCAT to choose another SMS storage group that has more space available and recover or reorganize the table space or index into the new SMS storage group.
- If the table space or index is DB2-managed but is not SMS-managed, rearrange the DB2 storage group volume list by shifting volumes with free space or by adding new volumes. Use the SQL statements ALTER STOGROUP ADD VOLUMES and ALTER STOGROUP REMOVE VOLUMES and then recover or reorganize the table space or index.
- If the table space or index is user-managed and not SMS-managed, delete and redefine the data set with more space and then recover or reorganize the table space or index.

Problem determination: Obtain an access method services LISTCAT for the named data set's VOLUME list for current extent space distribution of volumes.

See the Problem Determination section of message DSNP001I for more information.

DSNP031I *csect-name* **- CURRENT SPACE HAS** *cc*
EXTENTS FOR *data-set-name***. THE**
MAXIMUM EXTENTS IS *mm***.**
CONNECTION-ID=*xxxxxxxx***,**
CORRELATION-ID=*yyyyyyyyyyyy***,**
LUW-ID=*logical-unit-of-work-id=token*

Explanation: This message indicates that data set
data-set-name has almost reached the maximum number
of extents. The maximum number is indicated by *mm*.
You cannot add more volumes if the current volume
runs out of space.

The connection, correlation, and logical-unit-of-work
IDs identify the user or application that encountered
the problem. Correlation IDs beginning with *0nn*
(where *nn* is a number) identifies system agents.

A LUW-ID of an asterisk (*) indicates that the thread
does not have a LUW-ID. If the LUW-ID is not an
asterisk (*), the value is a *logical-unit-of-work-id* that is
assigned to the thread followed by an equals sign (=)
and a token value, which can be used in place of
logical-unit-of-work-id in any DB2 command that
accepts *logical-unit-of-work-id* as input.

Messages that have the same connection ID, correlation
ID, and LUW-ID relate to the same user. Including all
three pieces of information makes it possible to scan a
number of DSNP-prefixed messages and group
activities involving a given user.

Severity: 0 (informational)

System action: DB2 continues. If the table space or
index space is not able to extend when needed,
Message DSNP001I is issued and the current SQL
request fails.

Operator response: For more information, issue an
MVS WRITELOG operator command (for example, 'W
A') to obtain MVS SYSLOG output. The SYSLOG
sometimes contains VSAM or media manager messages
that might be helpful in determining why space could
not be allocated.

System programmer response: Based on the
information that might be available in the access
method services or VSAM messages preceding this one,
try one of the following actions:
• If you do not anticipate further extension of this
table space or index, ignore the message.

- If the table space or index is DB2-managed and has not yet reached the maximum number of extents, obtain an access method services LISTCAT for the named data set's to see how the extensions are distributed. Your options then include:
  - If there are multiple small extents, you can use the SQL statement ALTER TABLESPACE SECQTY to alter to a larger secondary space allocation quantity.
  - Otherwise, use the ALTER STOGROUP statement to add volumes to the storage group (up to 133 of them).
- If the table space or index is user-managed and has not yet reached the maximum number of extents, issue the access method services command ALTER ADDVOLUMES to provide additional volumes (up to 59 of them).
- If the data set is at the maximum number of extents, recover or reorganize the table space or index after taking one of the following actions:
  - Enlarge the primary or secondary space allocation quantity of the table space or index.
  - If the DB2 subsystem is running on z/OS V1.7 or later, and the data set is not shared with any z/OS systems at an earlier level, convert the data set to SMS-managed with the Extent Constraint Removal option set to YES in the SMS data class. If you do this, the maximum number of extents is 7257.

  These processes are described in Part 2 (Volume 1) of *DB2 Administration Guide*

Problem determination: Obtain an access method services LISTCAT for the named data set's VOLUME list for the extent space distribution.

See the Problem Determination section of message DSNP001I for more information.

Chapter 14. DSNQ... messages

DSNQ001I MQListener was invoked with no major function on the command line.

Explanation: The MQListener command line failed to specify a major function, such as help, run, add, remove, or show.

Severity: 8 (Error)

User response: Consult MQListener documentation and re-run MQListener with the function that was intended. For more information, run:

- db2mqln1/db2mqln2 help

DSNQ002I MQListener was invoked with an unknown command line parameter *parameter name*.

Explanation: The MQListener command line contained an unknown parameter *parameter name*.

Severity: 8 (Error)

User response: Consult MQListener documentation and re-run MQListener with the desired parameters. For more information, run

- db2mqln1/db2mqln2 help

DSNQ003I MQListener was invoked without the required command line parameter *parameter name*.

Explanation: The MQListener command line was missing the parameter *parameter name*, which was required.

Severity: 8 (Error)

User response: Consult MQListener documentation and re-run MQListener with the parameter that was required. For more information, run

- db2mqln1/db2mqln2 help

DSNQ004I MQListener was invoked with an illegal value for the *parameter name* command line parameter: *parameter value*.

Explanation: The MQListener command line contained an illegal value for the *parameter name* parameter. The illegal value was *parameter value*.

Severity: 8 (Error)

User response: Consult MQListener documentation and re-run MQListener with the desired parameters. For more information, run

- db2mqln1/db2mqln2 help

DSNQ010I MQListener encountered a database error accessing subsystem *subsystem name* during operation *operation name*: sqlstate = *sqlstate value* (sqlcode = *sqlcode value*).

Explanation: MQListener encountered a DB2 database error accessing subsystem *subsystem name* while executing operation *operation name*. The resulting sqlstate was *sqlstate value* (sqlcode *sqlcode value*).

Severity: 8 (Error)

User response: Consult MQListener documentation and ensure that MQListener is properly installed in the specified DB2 subsystem and access to MQListener packages and configuration tables ahs been granted to the user running MQListener.

DSNQ011I MQListener failed to make RRS connection to the subsystem *subsystem name* during operation *operation name*: reason code = *reason code value* (return code = *return code value*).

Explanation: MQListener encountered a DB2 database error while attempting to connect to subsystem *subsystem name* during operation *operation name*. The resulting reason code was *reason code value* (return code *return code value*).

Severity: 8 (Error)

User response: Ensure that the DB2 subsystem *subsystem name* is active and running.

DSNQ012I MQListener failed to terminate RRS connection from the subsystem *subsystem name* during operation *operation name*: reason code = *reason code value* (return code = *return code value*).

Explanation: MQListener encountered a DB2 database error while attempting to disconnect from subsystem *subsystem name* during operation *operation name*. The resulting reason code was *reason code value* (return code *return code value*).

Severity: 4 (Warning)

User response: Ensure that the DB2 subsystem *subsystem name* is active and running. Consult the DB2 documentation, regarding the exact meaning of the reason and return code.

© Copyright IBM Corp. 1983, 2008

DSNQ020I MQListener encountered a MQ error error executing operation *operation name* **on object** *object name* **in queue manager** *queue manager name***: reason code =** *reason code value***.**

Explanation: MQListener encountered a MQ error error while executing operation *operation name* on object *object name* in queue manager *queue manager name*. The resulting reason code was *reason code value*.

Severity: 8 (Error)

User response: Ensure that MQSeries is properly installed and configured, and accessible to the user running MQListener. Consult the MQSeries documentation for more information, particularly the Application Programming Reference for more information about the reason code.

DSNQ021I MQListener failed to connect to MQ queue manager *queue manager name***: reason code =** *reason code value***.**

Explanation: MQListener encountered a MQ error error while attempting to connect to MQ queue manager *queue manager name*. The resulting reason code was *reason code value*.

Severity: 8 (Error)

User response: Ensure that MQSeries is properly installed and configured, and accessible to the user running MQListener. Consult the MQSeries documentation for more information, particularly the Application Programming Reference for more information about the reason code.

DSNQ022I MQListener failed to disconnect from MQ queue manager *queue manager name***: reason code =** *reason code value***.**

Explanation: MQListener encountered a MQ error error while attempting to disconnect from MQ queue manager *queue manager name*. The resulting reason code was *reason code value*.

Severity: 4 (Warning)

User response: Ensure that MQSeries is properly installed and running. Consult the MQSeries documentation for more information, particularly the Application Programming Reference for more information about the reason code.

DSNQ030I MQListener encountered a DB2 database error attempting to add a new task to the MQListener configuration *configuration name* **in the subsystem** *subsystem name***. The input queue specified in the task was** *input queue name* **and the specified queue manager was** *queue manager name***. The resulting sqlstate was** *sqlstate value* **(sqlcode =** *sqlcode value* **).**

Explanation: MQListener encountered a DB2 database error while attempting to add a new task to the MQListener configuration *configuration name* in the subsystem *subsystem name*. The input queue specified in the task was *input queue name* and the specified queue manager was *queue manager name*. The resulting sqlstate was *sqlstate value* (sqlcode = *sqlcode value*).

Severity: 8 (Error)

User response: Consult MQListener documentation and ensure that MQListener is properly installed in the specified subsystemand and access to MQListener packages and configuration tables has been granted to the user running MQListener. Ensure that the combination of configuration name, input queue, and queue manager is unique among the MQListener configuration tasks in the subsystem. Use MQListener ″show″ command to inspect the configuration.

DSNQ040I MQListener encountered a DB2 database error retrieving configuration *configuration name* **from subsystem** *subsystem name* **during operation** *operation name***. The resulting sqlstate was** *sqlstate value* **(sqlcode =** *sqlcode value* **).**

Explanation: MQListener encountered a DB2 database error while retrieving configuration *configuration name* from subsystem *subsystem name* during operation *operation name*. The resulting sqlstate was *sqlstate value* (sqlcode= *sqlcode value*).

Severity: 8 (Error)

User response: Consult MQListener documentation and ensure that MQListener is properly installed in the specified subsystem and access to MQListener packages and configuration tables has been granted to the user running MQListener.

DSNQ060I MQListener encountered a DB2 database error while removing a task from configuration *configuration name* **in subsystem** *subsystem name***. The input queue specified was** *input queue name* **and the queue manager specified was** *queue manager name***. The resulting sqlstate was** *sqlstate value* **(sqlcode =** *sqlcode value* **).**

Explanation: MQListener encountered a DB2 database

error while removing a task from configuration *configuration name* in subsystem *subsystem name*. The input queue that was specified was *input queue name*, and the queue manager that was specified was *queue manager name*. The resulting sqlstate was *sqlstate value* (sqlcode = *sqlcode value*).

Severity: 8 (Error)

User response: Consult MQListener documentation and ensure that MQListener is properly installed in the specified subsystem. Also, ensure that access to MQListener packages and configuration tables has been granted to the user who is running MQListener.

DSNQ061I MQListener attempted to remove an unknown task from configuration *configuration name* **in subsystem** *subsystem name***. The input queue specified was** *input queue name* **and the queue manager specified was** *queue manager name***.**

Explanation: MQListener attempted to remove an unknown task from configuration *configuration name* in subsystem *subsystem name*. The input queue that was specified was *input queue name* and the queue manager that was specified was *queue manager name*.

Severity: 0 (Informational)

User response: Ensure that specified parameters are correct, and re-run MQListener. Use MQListener "show" command to inspect the configuration.

DSNQ070I MQListener has encountered a DB2 database error preparing to run the stored procedure *schema name***.** *procedure name* **in subsystem** *subsystem name***: sqlstate =** *sqlstate value* **(sqlcode =** *sqlcode value* **).**

Explanation: MQListener has encountered a DB2 database error while preparing to run the stored procedure *schema name*. *procedure name* in subsystem *subsystem name*. The resulting sqlstate was *sqlstate value* (sqlcode = *sqlcode value*).

Severity: 8 (Error)

User response: Ensure that MQListener configuration is correctly specified and that a stored procedure of the specified schema and name exists with a correct signature and is accessible to the user running MQListener.

DSNQ071I MQListener encountered an unsupported datatype in the signature of the stored procedure *schema name***.** *procedure name* **in the subsystem** *subsystem name***: datatype =** *datatype value***.**

Explanation: MQListener encountered an unsupported datatype in the signature of the stored procedure *schema name*. *procedure name* in the subsystem *subsystem name*. The datatype was *datatype value*.

Severity: 8 (Error)

User response: Consult the MQListener documentation and ensure that the stored procedure has the correct datatype.

DSNQ072I MQListener was unable to start a thread running a task specified in the configuration *configuration name* **by the input queue** *input queue name* **and queue manager** *queue manager name***. The ECF error code was** *error code value***.**

Explanation: MQListener was unable to start a thread running a task that was specified in the configuration *configuration name* by the input queue *input queue name* and queue manager *queue manager name*. The ECF error code was *error code value*.

Severity: 8 (Error)

User response: Ensure that MQListener configuration is correct, especially the -numInstances parameter, and that the MQListener process has sufficient system resources to execute the required number of simultaneous tasks.

DSNQ073I MQListener started a thread running the task specified in the configuration *configuration name* **by the input queue** *input queue name* **and queue manager** *queue manager name***.**

Explanation: MQListener started a thread running the task that was specified in the configuration *configuration name* by the input queue *input queue name* and queue manager *queue manager name*.

Severity: 0 (Informational)

User response: None. This is the formal notification of a thread start, expected during startup or restart.

DSNQ074I A thread running the task specified in the configuration *configuration name* **by the input queue** *input queue name* **and queue manager** *queue manager name* **has exited.**

Explanation: A thread running the task specified in the configuration *configuration name* by the input queue *input queue name* and queue manager *queue manager name* has exited.

Severity: 0 (Informational)

User response: None. This is the formal notification of a thread exit, expected during shutdown or restart.

DSNQ075I MQListener encountered a DB2 database error while running the stored procedure *schema name*. *procedure name* **in subsystem** *subsystem name* **during operation** *operation name*. **The resulting sqlstate was** *sqlstate value* **(sqlcode =** *sqlcode value* **).**

Explanation: MQListener encountered a DB2 database error while running the stored procedure *schema name*. *procedure name* in subsystem *subsystem name* during operation *operation name*. The resulting sqlstate was *sqlstate value* (sqlcode = *sqlcode value*).

Severity: 8 (Error)

User response: Consult MQListener documentation and ensure that MQListener is properly installed in the specified subsystem and access to MQListener packages and configuration tables has been granted to the user running MQListener. Ensure that the stored procedure executes correctly.

DSNQ080I MQListener received an unknown message on admin queue *admin queue name* **in queue manager** *queue manager name*: *message*.

Explanation: MQListener received an unknown message on admin queue *admin queue name* in queue manager *queue manager name*: *message*.

Severity: 4 (Warning)

User response: Ensure that MQSeries is correctly installed and configured, that MQListener is being run with the proper -adminQMgr and -adminQueue parameter values, and that the MQListener ″admin″ command is also using the proper -adminQMgr and -adminQueue or -adminQueueList parameter values. If used, check that the -adminQueueList specifies the proper queue names. Check that the admin queue is not being used by another application.

DSNQ081I MQListener received a shutdown message on admin queue *admin queue name*, **queue manager** *queue manager name*.

Explanation: MQListener received a shutdown message on admin queue *admin queue name*, queue manager *queue manager name*.

Severity: 0 (Informational)

User response: None. This is a formal notification on receiving a shutdown message.

DSNQ082I MQListener received a restart message on admin queue *admin queue name*, **queue manager** *queue manager name*.

Explanation: MQListener received a restart message on admin queue *admin queue name*, queue manager *queue manager name*.

Severity: 0 (Informational)

User response: None. This is a formal notification on receiving a restart message.

Chapter 15. DSNR... messages

The following information messages indicate the progress of DB2 through a restart process.

DSNR001I RESTART INITIATED

Explanation: This message delimits the beginning of the restart process within startup. The phases of restart are about to begin. These phases are necessary to restore the operational environment to that which existed at the time of the previous termination and to perform any recovery actions that may be necessary to return DB2-managed resources to a consistent state.

DSNR002I RESTART COMPLETED

Explanation: This message delimits the completion of the restart process within startup.

DSNR003I RESTART...PRIOR CHECKPOINT RBA=*xxxxxxxxxxxx*

Explanation: The message indicates the first phase of the restart process is in progress and identifies (as 'xxxxxxxxxxxx') the log positioning RBA of the checkpoint from which the restart process will obtain its initial recovery information.

DSNR004I RESTART...UR STATUS COUNTS IN COMMIT=*nnnn*, INDOUBT=*nnnn*, INFLIGHT=*nnnn*, IN ABORT=*nnnn*, POSTPONED ABORT=*nnnn*

Explanation: This message indicates the completion of the first phase of the restart process. The counts indicate the number of units of work whose execution state during a previous DB2 termination was such that (to ensure DB2 resource consistency) some recovery action must be performed during this restart process or (by the user) upon completion of the restart process. The counts of IN COMMIT, INDOUBT, INFLIGHT and IN ABORT may provide an indication of the time required to perform the remaining two phases of restart (forward and backward recovery). The counts of POSTPONED ABORT URs may provide an indication of the time required to perform -RECOVER POSTPONED processing.

The IN COMMIT count specifies the number that had started but not completed phase 2 of the commit process. These must undergo forward recovery to complete the commit process.

The INDOUBT count specifies the number that were interrupted between phase 1 and phase 2 of the commit process. These must undergo forward recovery to ensure resources modified by them are unavailable until their indoubt status is resolved.

The INFLIGHT count specifies the number that neither completed phase 1 of the commit process nor began the process of backing out. These must undergo backward recovery to restore resources modified by them to their previous consistent state.

The IN ABORT count specifies the number that were in the process of backing out. These must undergo backward recovery to restore resources modified by them to their previous consistent state.

The POSTPONED ABORT count specifies the number whose backout was postponed in an earlier restart process and not attended to prior to this restart process. These must undergo backward recovery after completion of restart, via the -RECOVER POSTPONED command, to restore resources modified by them to their previous consistent state.

This message is issued by the following CSECT: DSNRRCSR

DSNR005I RESTART...COUNTS AFTER FORWARD RECOVERY IN COMMIT=*nnnn*, INDOUBT=*nnnn*

Explanation: The message indicates the completion of the forward recovery restart phase. The counts indicate the number of units of work whose recovery actions could not be completed during the phase. Typically, those in an IN COMMIT state remain because the recovery actions of some subcomponents have not been completed. Those units of work in an INDOUBT state will remain until connection is made with the subsystem that acts as their commit coordinator.

Operator response: No action is required unless the conditions persist beyond some installation-defined period of time. Recovery action will be initiated when the resource is brought online. Indoubt resolution will be initiated as part of the process of reconnecting the subsystems.

DSNR006I RESTART...COUNTS AFTER BACKWARD RECOVERY INFLIGHT=*nnnn*, IN ABORT=*nnnn*, POSTPONED ABORT=*nnnn*

Explanation: The message indicates the completion of the backward recovery restart phase. The counts indicate the number of units of work whose recovery actions could not be completed during the phase.

© Copyright IBM Corp. 1983, 2008

When INFLIGHT or IN ABORT URs do not complete backout during restart, they are converted to POSTPONED ABORT status, so the INFLIGHT and IN ABORT counts should be zero. The POSTPONED ABORT count indicates the number of URs for which backout work still exits (from this or a previous restart).

Backout work for the POSTPONED ABORT URs should be completed by the user (via the -RECOVER POSTPONED command), once DB2 restart has completed.

This message is issued by the following CSECT: DSNRRAUB

Operator response: No action is required unless the condition persists beyond some installation-defined period of time. Recovery action will be initiated when the resource collection is brought online.

DSNR007I RESTART...STATUS TABLE

Explanation: This message precedes a table showing the status of units of recovery (URs) after each restart phase. The message and the table will accompany the DSNR004I, DSNR005I, or DSNR006I message after each nested phase. At the end of the current status rebuild phase, it shows the status of any URs that require processing. At the end of the forward recovery and backoutphases, it shows the status of only those URs which needed processing but were not processed. The table helps to identify the URs that were active when DB2 came down, and to determine the log scope required to bring DB2 up.

The following list describes each column of the table:

T Connection type. The values could be:

- **B** Batch: a batch program or QMF user.
- **I** Internal: a system service being performed on some user's behalf.
- **S** Subsystem: work that originated from an attached subsystem, either CICS or IMS/VS.

CONID
Connection identifier for related URs. Batch connections are not related to any other connection. Subsystem connections with the same identifier indicate URs that originated from the same subsystem.

CORRID
Correlation identifier; unique within a subsystem connection, it relates a UR to a particular CICS or IMS/VS transaction.

AUTHORIZATION ID
Authorization identifier of the user for whom this UR was created.

PLAN Name of the plan allocated to the UR.

S Restart status of the UR. When DB2 came down, the UR was in one of these situations:

- **A** INABORT: The UR was in the 'must-complete' phase of abort, and is yet to be completed.
- **C** INCOMMIT: The UR was in the 'must-complete' phase of commit, and is yet to be completed.
- **D** INDOUBT: The UR had completed the first phase of commit, but DB2 had not received the second phase instruction. The UR must be remembered so it can be resolved when the owning subsystem reattaches.
- **F** INFLIGHT: The UR had not completed the first phase of commit, and will be backed out.
- **P** POSTPONED ABORT: The UR has not yet completed backout. Backout must be completed outside of DB2 restart by means of the -RECOVER POSTPONED command.

URID UR identifier, the log RBA of the beginning of this unit of work. It is the earliest RBA required to process the UR during Restart.

DAY The day the UR was created, in the format 'ddd'. It is approximately the time of the first SQL statement of the application or the first SQL statement following a commit point.

TIME The time the UR was created, in the format 'hh mm ss '. It is approximately the time of the first SQL statement of the application or the first SQL statement following a commit point.

DSNR009I STORAGE IS NOT AVAILABLE FOR THE UR DISPLAY TABLE, THE SIZE REQUESTED = *xxxx*, STORAGE MANAGER REASON CODE = *yyyyyyyy*

Explanation: There was not enough storage available during the creation of the recoverable UR (RURE) display table (DSNDRUDT).

System action: The system continues to restart but the RESTART... STATUS TABLE is not displayed.

System programmer response: Increase the region size of the *ssnm*MSTR address space before restarting DB2 again.

Operator response: Save the console output and inform the system programmer.

Problem determination: The size requested is approximately 110 bytes for each unit of recovery (UR). See the message DSNR004I to determine the total number of URs to process. Then get the 00E2xxxx

reason code from the DSNR009I header. Use these two numbers to determine the reason for the shortage.

DSNR010I AN ERROR OCCURRED IN THE RURE STATUS TABLE SORT/TRANSLATE MODULE (DSNRUSOR), THE ERROR LOCATION CODE = *xxxx*

Explanation: There was an error in processing the RURE display table. One of these ERROR LOCATION CODES tells what function the module was attempting to perform:

X'0001' Trace entry to the module
X'0002' Sort the recoverable UR display table
X'0003' Translate the recoverable UR table
X'0004' Convert the store clock value for display
X'0005' Trace exit from the module

System action: The system continues to restart but the RESTART... STATUS TABLE is not displayed.

System programmer response: Use the error code in the message to fill in the problem description.

Operator response: Save the console output and inform the system programmer.

Problem determination: The ERROR CODE tells the problem area and should be used in the problem description. A record was added to the SYS1.LOGREC that contains the PSW and module at the time of the error. This error indicates an application program check of some type.

DSNR011I AN ERROR OCCURRED IN THE RURE STATUS TABLE DISPLAY MODULE (DSNRUDIS), THE ERROR LOCATION CODE = *xxxx*

Explanation: There was an error in processing the RURE display table. One of these ERROR LOCATION CODES tells what function the module was attempting to perform:

X'0001' Trace entry to the module
X'0002' Build header for the display table
X'0003' Build entries for the display table
X'0004' Call DSNRWTO to display the table
X'0005' Trace exit from the module
X'0006' Get message text

System action: The system continues to restart but the RESTART... STATUS TABLE is not displayed.

System programmer response: Use the error code in the message to fill in the problem description.

Operator response: Save the console output and inform the system programmer.

Problem determination: The ERROR CODE describes the problem area and should be used in the problem description. A record was added to the SYS1.LOGREC containing the value DSNRUDIS, and the PSW and CSECT at the time of the error. This error indicates an application program check of some type.

Error codes other than X'0006' indicate an application program check of some type. Code X'0006' indicates an error attempting to locate the table text. The trace entry for DSNRUDIS contains the return and reason codes from the Get Text request. That information should be used to fill in the problem description.

DSNR014I EXCLUDED RURE TABLE

Explanation: This message contains a list of RUREs which are being bypassed because of the conditional restart. Either they were to be processed in a phase that was bypassed or they were outside the log range.

The following list describes each column of the table:

T Connection type. The values could be:

B Batch: a batch program or QMF user.

I Internal: a system service being performed on some user's behalf.

S Subsystem: work that originated from an attached subsystem, either CICS or IMS/VS.

CONID
Connection identifier for related URs. Batch connections are not related to any other connection. Subsystem connections with the same identifier indicate URs that originated from the same subsystem.

CORRID
Correlation identifier; unique within a subsystem connection, it relates a UR to a particular CICS or IMS/VS transaction.

AUTHORIZATION ID
Authorization identifier of the user for whom this UR was created.

PLAN Name of the plan allocated to the UR.

S Restart status of the UR. When DB2 came down, the UR was in one of these situations:

A INABORT: The UR was in the 'must-complete' phase of abort, and is yet to be completed.

C INCOMMIT: The UR was in the 'must-complete' phase of commit, and is yet to be completed.

D INDOUBT: The UR had completed the first phase of commit, but DB2 had not received the second phase instruction. The UR must be remembered so it can be resolved when the owning subsystem reattaches.

F INFLIGHT: The UR had not completed the first phase of commit, and will be backed out.

P POSTPONED ABORT: The UR has not yet completed backout. Backout must be completed outside of DB2 restart by means of the -RECOVER POSTPONED command.

URID UR identifier, the log RBA of the beginning of this unit of work. It is the earliest RBA required to process the UR during Restart.

DAY The day the UR was created, in the format 'ddd'. It is approximately the time of the first SQL statement of the application or the first SQL statement following a commit point.

TIME The time the UR was created, in the format 'hh mm ss '. It is approximately the time of the first SQL statement of the application or the first SQL statement following a commit point.

System action: Restart continues.

DSNR015I THE CHKPTRBA *xxxxxxxxxxxx* **SPECIFIED IN THE CRCR IS INVALID**

Explanation: The checkpoint RBA specified in the conditional restart control record (CRCR) is either not available or is not a begin checkpoint record.

System action: Restart is abended with reason code '00D99001'.

Operator response: Inform the system programmer.

System programmer response: By specifying the STARTRBA and/or ENDRBA on the CRESTART control statement, change log inventory (DSNJU003) utility selects an appropriate RBA.

Problem determination: Print the complete checkpoint records queue by using the print log map (DSNJU004) utility. Select a checkpoint RBA from this queue or let the change log inventory (DSNJU003) utility make the selection.

DSNR016I STARTRBA *xxxxxxxxxxxx* **VALIDATION FAILED**

Explanation: A scan from the STARTRBA failed to find a complete log record. Either the entire log was scanned and no complete record was found, or the log manager found an invalid log record. The log manager sends the error messages to the console indicating a bad log record.

System action: Restart abends.

Operator response: Record the log manager message numbers and inform the system programmer.

System programmer response: Use DSN1LOGP to determine a STARTRBA to use for restart. The recovery manager uses the STARTRBA as a starting point for its search of the beginning of a valid log record.

DSNR017I RESTART IS BEING TERMINATED BECAUSE OF CSRONLY SPECIFICATION IN CRCR

Explanation: The restart procedure only completed the first phase of startup. The status displayed during this partial restart can be used to build the conditional restart control record (CRCR).

System action: Restart is abended.

System programmer response: To prevent the restart control record from being reused during the next restart attempt, the current restart control record must be canceled. Use the change log inventory (DSNJU003) CRESTART CANCEL option, or create a new restart control record. The newly created restart control record will automatically cancel the previous record.

DSNR018I *csect-name* **RESTART...BACKWARD RECOVERY PROCESSED FROM RBA** *ffffffffffff* **TO RBA** *tttttttttttt*

Explanation: This message indicates the range of log processed during the backward phase of RESTART when postponed backout processing has been requested. Processing began with the LR of RBA *ffffffffffff* and proceeded backwards to the LR of RBA *tttttttttttt*.

System action: This message is followed by messages DSNR006I and DSNR007I. The former tells how many URs had some backout work postponed. The latter, identifies the postponed abort URs. The amount of log that need to be processed to complete backout processing for the postponed abort URs can be gauged by comparing the RBA reached during RESTART's backward processing (RBA *tttttttttttt*), and the URIDs of the postponed abort URs.

DSNR020I *csect-name* **START MEMBER** *member*, **OR REPLY 'NO' OR 'QUIESCED'**

Explanation: During a group restart, the group member issuing this message attempted to access the BSDS for member *member*, and either was not able to access the BSDS, or discovered that the BSDS was modified by the change log inventory (DSNJU003) utility since that member was last started. In either case, this DB2 is not able to perform peer restart for member *member*.

For group restart to continue, the identified member must be started.

If it is not possible to start the other member and you want to terminate this group restart, you can reply 'NO' to this message. Replying 'NO' to this message causes this DB2 to terminate with a reason code of 00D900E1. You then have to reply 'NO' to this message for all remaining members participating in this group restart.

If you are sure that the identified member was quiesced the last time it was started, that is, it was

stopped with a -STOP DB2 MODE(QUIESCE) command and it stopped with no indoubt units of recovery and no distributed resynchronization responsibility, then you can reply 'QUIESCED' to this message. Replying 'QUIESCED' to this message causes message DSNR030I to be issued as a reminder that group restart will continue **without using the log** from the identified member.

System action: This group member waits for the identified member to be started or for a valid reply to this message.

- If the identified member is started, group restart continues.
- If the reply is 'NO', this DB2 terminates with a reason code of 00D900E1.
- If the reply is 'QUIESCED', group restart continues **without using the log** from the identified member.

DSNR021I *csect-name* **DB2 SUBSYSTEM MUST PERFORM GROUP RESTART FOR PEER MEMBERS**

Explanation: This DB2 subsystem must perform the group restart phase to recover either the shared communications area (SCA) or the retained locks for nonstarting members in the data sharing group.

This message marks the beginning of group restart processing for peer members. Messages DSNR023I and DSNR024I or messages DSNR025I and DSNR026I are issued for each peer that is processed by this DB2 subsystem. Message DSNR022I marks the end of group restart processing for peer members.

This message is issued before this DB2 member enters the group restart process. The group restart process is synchronized among all group members that are starting and therefore participating in this group restart. After this DB2 member synchronizes with other members, it is possible that no peer processing will be necessary; in that case message DSNR022I immediately follows this message.

If it is necessary to recover both the SCA and retained locks, this message is issued twice; once for SCA recovery and again for the recovery of the retained locks.

System action: The subsystem begins group restart processing.

DSNR022I *csect-name* **DB2 SUBSYSTEM HAS COMPLETED GROUP RESTART FOR PEER MEMBERS**

Explanation: This DB2 subsystem, which previously issued message DSNR021I, completed the group restart phase to recover the shared communications area (SCA) or retained locks for nonstarting DB2 members in the data sharing group.

System action: The subsystem continues with the next restart phase of its own subsystem.

DSNR023I *csect-name* **GROUP RESTART INITIATED TO RECOVER THE SCA FOR GROUP MEMBER** *membname*

Explanation: The DB2 subsystem that issued this message is performing the group restart phase to recover the shared communications area (SCA) for nonstarting member *membname* of the data sharing group.

This message is followed by message DSNR024I after the group restart phase to recover the SCA for the identified member is complete.

System action: The subsystem continues the group restart process.

DSNR024I *csect-name* **GROUP RESTART COMPLETED TO RECOVER THE SCA FOR GROUP MEMBER** *membname*

Explanation: The DB2 subsystem that issued this message completed the group restart phase to recover the shared communications area (SCA) for member *membname* of the data sharing group.

This message is issued only if messages DSNR021I and DSNR023I were issued.

System action: The subsystem continues the group restart process.

DSNR025I *csect-name* **GROUP RESTART INITIATED TO RECOVER RETAINED LOCKS FOR GROUP MEMBER** *membname*

Explanation: The DB2 subsystem that issued this message is performing the group restart phase to recover the retained locks for nonstarting member *membname* of the data sharing group.

This message is followed by message DSNR026I after the group restart phase to recover the retained locks for the identified member is complete.

System action: The subsystem continues the group restart process.

User response: Restart the failed member to release the retained locks.

DSNR026I *csect-name* **GROUP RESTART COMPLETED TO RECOVER RETAINED LOCKS FOR GROUP MEMBER** *membname*

Explanation: The DB2 subsystem that issued this message completed the group restart phase to recover the retained locks for member *membname* of the data sharing group.

This message is issued only if messages DSNR021I and DSNR025I were issued.

System action: The subsystem continues the group restart process.

DSNR030I *csect-name* **WILL CONTINUE WITHOUT THE** *member* **MEMBER'S LOG, REPLY 'YES' OR 'NO'**

Explanation: The reply to message DSNR020I was 'QUIESCED'. This message is issued as a reminder that group restart will continue **without using the log** from the identified member.

If you are sure that the identified member was quiesced the last time it was started, that is, it was stopped with a -STOP DB2 MODE(QUIESCE) command and it stopped with no indoubt units of recovery and no distributed resynchronization responsibility, then reply 'YES'; otherwise, reply 'NO'.

System action: This group member waits for a valid reply to this message.

- If the reply is 'YES', group restart continues **without using the log** from the identified member.
- If the reply is 'NO', message DSNR020I is reissued to allow you to enter a different response.

DSNR031I *csect-name* **PROCESSING LOG RECORD AT RBA** *rba1* **TO RBA** *rba2*

Explanation: This message is produced periodically during the forward and backward processing phases of restart. The RBAs presented in this message can be used to monitor the progress of the phases of restart.

Forward processing occurs between messages DSNR004I and DSNR005I. During forward processing, the log is read in a forward direction.

Backward processing occurs between messages DSNR005I and DSNR006I. During backward processing, the log is read in a backward direction and will stop reading at the RBA matching the URID of the oldest in-flight or in-abort UR.

rba1 is the RBA of the log record currently being processed. *rba2* is the RBA of the last log record that will be read in this phase of restart.

System action: DB2 restart continues.

User response: This is an information only message.

DSNR034E *csect-name* **THERE CANNOT BE MORE THAN TWO RELEASE LEVELS OF DB2 RUNNING IN THE GROUP**

Explanation: You attempted to start a DB2 that was at a release level that cannot coexist in the data sharing group. There are already two different releases of DB2 currently running in the data sharing group.

System action: DB2 terminates.

System programmer response: If you are sure that the startup procedure is pointing to the correct set of load libraries, you must stop any DB2 subsystems that cannot coexist with the one you are attempting to start.

See *DB2 Data Sharing: Planning and Administration* for more information about release coexistence in a data sharing group.

DSNR035I *csect-name* **WARNING - UNCOMMITTED UR AFTER** *nn* **CHECKPOINTS - CORRELATION NAME =** *xxxxxxxxxxxx* **CONNECTION ID =** *yyyyyyyy* **LUWID =** *logical-unit-of-work-id=token* **PLAN NAME =** *xxxxxxxx* **AUTHID =** *xxxxxxx* **END USER ID =** *xxxxxxxx* **TRANSACTION NAME =** *xxxxxxxx* **WORKSTATION NAME =** *xxxxxxxx*

Explanation: This message indicates that during checkpoint, DB2 encountered an uncommitted UR that has an INFLIGHT status.

The value *nn* is cumulative, and it indicates the number of checkpoints taken since the beginning of the UR. CORRELATION name, CONNECTION ID, and LUWID together identify a thread associated with the UR. If the LUWID is an '*', it indicates that the thread originated at this site. *token* is a unique token number associated with the LUWID. PLAN NAME and AUTHID further identify the thread associated with the UR. If the thread was created with client End User information, the End User's ID, TRANSACTION NAME, and WORKSTATION NAME will be displayed. Otherwise, these fields will contain an '*'.

System action: DB2 continues processing. If statistics class 3 is turned on, IFCID 0313 is written.

System programmer response: Consult with the application programmer to determine if this is a problem UR. See Part 4 (Volume 1) of *DB2 Administration Guide* for more information about problems caused by uncommitted URs. If the UR is caused by an application program, you can use CANCEL THREAD to delete the UR, if necessary. If an uncommitted UR is deleted, DB2 will undo the changes. The amount of time required for this process depends on the amount of work done by the UR.

Programmer response: Ensure that the application commits frequently enough, or consult with your DB2 administrator about decreasing the frequency of the check.

DSNR036I *csect-name* **UNRESOLVED UR ENCOUNTERED DURING CHECKPOINT - CORRELATION NAME =** *xxxxxxxxxxx* **CONNECTION ID =** *yyyyyyyy* **LUWID =** *logical-unit-of-work-id* **PLAN NAME =** *xxxxxxxx* **AUTHID =** *xxxxxxxx*

Explanation: DB2 has detected an INDOUBT unit of recovery (UR) during a checkpoint. The purpose of this message is to remind the operator that an INDOUBT UR or POSTPONED ABORT UR exists.

CORRELATION name, CONNECTION ID, and LUWID together identify a thread associated with the UR. If the LUWID is an '*', it indicates that the thread originated at this site. PLAN NAME and AUTHID further identify the thread associated with the indoubt UR.

System action: DB2 continues processing. If statistics class 3 is turned on, IFCID 0313 is written.

System programmer response: Consult with the application programmer to determine whether this is a problem UR. See Part 4 (Volume 1) of *DB2 Administration Guide* for more information about resolving INDOUBT URs.

DSNR037I *csect-name* **WARNING - THE TCP/IP DRDA PORT NUMBER IS CHANGED FROM** *port_1* **TO** *port_2*

Explanation: The TCP/IP DRDA port number specified in the DB2 BSDS is different from the value used the last time a member of the DB2 data sharing group started. All members of a DB2 data sharing group must use the same TCP/IP DRDA port number.

port_1 The TCP/IP DRDA port number that was obtained from the DB2 BSDS during DB2 startup.

port_2 The TCP/IP DRDA port number that was obtained from the coupling facility during DB2 startup. This is the TCP/IP DRDA port number that was used the last time a member of the data sharing group was started.

System action: DB2 accepts the new TCP/IP DRDA port number, and DB2 startup continues.

Operator response: Contact the system programmer.

System programmer response: If the TCP/IP DRDA port number was changed inadvertently, stop DB2 immediately. Use the change log inventory utility (DSNJU003) to correct the PORT value, and restart DB2.

If this message reflects a change that you want to make to DB2's DRDA port number, no action is required. The DRDA port number stored in the coupling facility is updated, and this new port number will be used when DB2 validates each member's DRDA port number.

DSNR038I *csect-name* **WARNING - THE TCP/IP RESYNCHRONIZATION PORT NUMBER IS CHANGED FROM** *port_1* **TO** *port_2*

Explanation: The TCP/IP resynchronization port number obtained from the DDF record of the BSDS is different than the value specified the last time DB2 started.

Changing the TCP/IP resynchronization port number can cause problems during distributed 2-phase commit processing. Remote DRDA partners record DB2's resynchronization port number in their recovery logs. The resynchronization port number is used to reconnect to the appropriate DB2 member when communication failures occur during 2-phase commit processing. If a DB2 member changes its resynchronization port number, the remote DRDA partners may be unable to resolve indoubt units of work.

port_1 The TCP/IP resynchronization port number that was obtained from the DB2 BSDS during DB2 startup.

port_2 The TCP/IP resynchronization port number that was obtained from the coupling facility during DB2 startup. This is the TCP/IP resynchronization port number that was used the last time this member was started.

System action: DB2 accepts the new TCP/IP resynchronization port number, and DB2 startup continues.

Operator response: Contact the system programmer.

System programmer response: If the DB2 resynchronization port number was changed inadvertently, stop DB2 immediately. Use the change log inventory utility (DSNJU003) to correct the RESPORT value, and restart DB2.

If this message reflects a change that you want to make to DB2's resynchronization port number, no action is required. The resynchronization port number stored in the coupling facility for this member is updated, and this new port number will be used when DB2 validates the resynchronization port number for other DB2 members.

DSNR039I *csect-name* **WARNING - TCP/IP RESYNCHRONIZATION PORT NUMBER** *port_1* **IS THE SAME AS MEMBER** *member_name*

Explanation: The TCP/IP resynchronization port of the starting DB2 is the same as the member indicated by *member_name*.

port_1 The TCP/IP resynchronization port number that was obtained from the DB2 BSDS during DB2 startup.

member_name
The member name of the DB2 data sharing member that used the same resynchronization port number.

System action: The TCP/IP port number specified in the BSDS is accepted, and DB2 startup continues.

Operator response: Contact the system programmer.

System programmer response: Each member of the DB2 data sharing group should have a unique TCP/IP resynchronization port number. Use the change log inventory utility (DSNJU003) to specify a unique RESPORT value for this DB2 member and the DB2 member identified by *member_name*.

DSNR040I *csect-name* **TRACKER RESTART ENDLRSN** *lrsn* **DOES NOT MATCH ENDLRSN** *lrsn* **ASSOCIATED WITH THE DATA SHARING GROUP**

Explanation: In a data sharing environment, a TRKRSITE DB2 will not restart if it detects inconsistencies in the reported values for ENDLRSN. The ENDLRSN specified on the Conditional Restart Control Record does not match the ENDLRSN in the system communications area (SCA) structure for the data sharing group.

System action: DB2 startup terminates.

Operator response: Collect the SYS1.LOGREC and SVC dump, and contact the system programmer.

System programmer response: If the ENDLRSN value for this member is incorrect, take the following actions:

- Restore the BSDS and Logs for his Tracker DB2 from the primary DB2
- Ensure the Conditional Restart Control Record is correct
- Restart DB2

If the ENDLRSN value in the SCA is incorrect, delete the SCA structure and restart all members of the TRKRSITE data sharing group with the same ENDLRSN.

DSNR042I *csect-name* **WARNING - UR ROLLBACK HAS BEEN CANCELED AND IS INCOMPLETE FOR CORRELATION NAME =** *corrid* **CONNECTION ID =** *connid* **AUTHID =** *authid* **PLAN NAME =** *plan-name* **URID =** *urid*

Explanation: DB2 issues this message when rollback for the indicated thread has been canceled by either the CANCEL THREAD NOBACKOUT command or the RECOVER POSTPONED CANCEL command.

System action: Rollback processing for the thread stops and the unit of recovery is logged as canceled and complete.

User response: Examine the console for messages indicating what objects are being marked as REFP,RECP; those objects will need to be recovered.

DSNR043E *csect-name* **DB2 IS STARTING WITH AN UNSUPPORTED CODE LEVEL. CURRENT LEVEL:** *current-code-level* **VALID LEVEL(S):** *valid-code-levels*

Explanation: The DB2 that is starting is not at the proper code level. Some required maintenance, such as a fallback or coexistence SPE, has not been applied. The level information is different for each SPE level and each release. This information is for the use of your IBM service representative.

This message can occur in the following situations:

- The DB2 that is starting has completed a fallback from a later release and does not have the proper fallback SPE applied.
- The DB2 that is starting does not have the proper coexistence SPE to start in a data sharing group, which is at a later release level.
- The DB2 that is starting is not at a release level that supports coexistence with the data sharing group.

System action: DB2 terminates with reason code 00D900FA.

Operator response: Save the console output and inform the system programmer.

System programmer response: Check the DB2 Program Directory and Preventive Service Planning updates (PSP) to determine what SPE you need.

Verify that:

- Coexistence is supported between the releases you are trying to run.
- The DB2 startup procedure points to the correct set of load libraries.
- The correct SPE has been applied.

Problem determination: Collect the following diagnostic items listed in Appendix A, “Problem determination,” on page 659: 1, 2, 16, 69.

DSNR044E *csect-name* **MEMBER** *member-name* **IS STARTED AT A LEVEL OF CODE** *started-code-level* **WHICH IS NOT COMPATIBLE WITH THE STARTING MEMBER. VALID LEVEL(S)** *valid-code-levels*

Explanation: The code levels are not compatible between the DB2 member that has already started and the DB2 member that is starting. You need to apply the correct SPE to the DB2 member that is at the earlier level.

The level information is different for each SPE level and each release. This information is for the use of

your IBM service representative.

This message occurs when a starting DB2 member is at a later release level that cannot coexist with an existing DB2 member.

System action: After detecting the first member with which it cannot coexist, DB2 terminates with reason code 00D900FA.

Operator response: Save the console output and inform the system programmer.

System programmer response: Check the Program Directory and Preventive Service Planning updates (PSP) to ensure that all active members in the group have the proper SPE applied.

Verify that:

- Coexistence is supported between the releases you are trying to run.
- The DB2 startup procedure points to the correct set of load libraries.
- The correct SPE has been applied.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 16, 69.

DSNR045I *csect-name* **DB2 SUBSYSTEM IS STARTING AND IT WAS NOT STARTED IN A PREVIOUS RELEASE WITH THE FALLBACK SPE APPLIED. FALLBACK SPE APAR:** *apar-number* **NEW RELEASE LEVEL:** *release* **KNOWN LEVEL(S):** *level*

Explanation: DB2 was not started with the FALLBACK SPE active before an attempt was made to start DB2 in a new release. The FALLBACK SPE must be applied, and the system must be started before DB2 can be started in a new release.

release level
The release and level are differnt for each release, and are for the use of the IBM service representative.

apar-number
The apar-number is the FALLBACK SPE that is missing.

System action: DB2 terminates with reason code 00D96001.

User response: Ensure that the FALLBACK SPE is applied, and DB2 is started before starting DB2 in a new release.

DSNR046I *csect-name* **INCOMPLETE UNITS OF RECOVERY EXIST FOR** *subsys*

Explanation: This message is issued during DB2 shutdown if any incomplete units of recovery exist for the referenced member of a data sharing group. The message indicates that retained locks will be held by subsystem *subsys*.

System action: DB2 shutdown continues normally.

User response: Restart DB2 and resolve the incomplete units of recovery to release the retained locks held by subsystem *subsys*.

DSNR047I *csect-name* **POSTPONED ABORT BACKOUT PROCESSING LOG RECORD AT RBA** *rba1* **TO RBA** *rba2*

Explanation: This message is produced periodically during the backout processing of postponed abort units of recovery, initiated by the -RECOVER POSTPONED command. The RBAs in this message can be used to monitor the backout progress.

During the backout process, DB2 reads the log in a backward direction and stops reading the log at the RBA matching the URID of the oldest postponed-abort UR *rba2*. *rba1* is the RBA of the log record currently being processed.

System action: Backout processing continues.

System programmer response: This is an information-only message. If the message continues to be issued and *rba1* remains the same, make sure there are no outstanding archive tape mounts that may be delaying the log read process. Otherwise, obtain a dump of the DB2 address spaces for problem determination.

The -DISPLAY THREAD TYPE(POSTPONED) command can be issued to display information about the postponed abort units of recovery. It is also possible to stop the backout process with the -RECOVER POSTPONED CANCEL command, but this will leave objects in an inconsistent state. See the *DB2 Command Reference* for more information regarding the CANCEL option of the -RECOVER POSTPONED command.

DSNR048I *csect-name* **UR BACKOUT PROCESSING LOG RECORD AT RBA** *rba1* **TO RBA** *rba2* **FOR CORRELATION NAME =** *xxxxxxxxxxxx* **CONNECTION ID =** *yyyyyyyy* **LUWID =** *logical-unit-of-work-ID*=*token* **PLAN NAME =** *xxxxxxxx* **AUTHID =** *xxxxxxxx* **END USER ID =** *xxxxxxxx* **TRANSACTION NAME =** *xxxxxxxx* **WORKSTATION NAME =** *xxxxxxxx*

Explanation: This message is produced periodically during the backout process of an in-abort unit of recovery. The RBAs in this message can be used to monitor the backout progress. Together, CORRELATION name, CONNECTION ID, and LUWID identify a thread associated with the UR. If the LUWID is '*', the thread originated at this site. *token* is a unique token number associated with the LUWID. PLAN NAME and AUTHID further identify the thread

associated with the UR. If the thread was created with client end user information, the end user's USERID, TRANSACTION NAME, and WORKSTATION NAME are displayed. Otherwise, these fields contain an '*'.

During the backout process, DB2 reads the log in a backward direction and stops reading the log at the RBA matching the URID of the in-abort UR *rba2*. *rba1* is the RBA of the log record currently being processed.

System action: Backout processing continues.

System programmer response: This is an information-only message. If the message continues to be issued and *rba1* remains the same, make sure there are no outstanding archive tape mounts that might be delaying the log read process. Otherwise, obtain a dump of the DB2 address spaces for problem determination.

It might be possible to stop the backout process with the NOBACKOUT option of the -CANCEL THREAD command, but this leaves objects in an inconsistent state. See the *DB2 Command Reference* for more information regarding the NOBACKOUT option of the -CANCEL THREAD command.

DSNR050I *csect-name* **DB2 STARTED IN SYSTEM RECOVER PENDING MODE**

Explanation: DB2 has started in System Recover Pending mode in preparation for the RESTORE SYSTEM utility. No database update activity is allowed, and DB2 is in restricted access mode.

System action: DB2 startup continues.

DSNR051I *csect-name* **SYSPITR RESTART ENDLRSN** *lrsn* **DOES NOT MATCH ENDLRSN** *lrsn* **ASSOCIATED WITH THE DATA SHARING GROUP**

Explanation: For a System Level Recovery restart, all members of the data sharing group must truncate their logs with the same LRSN value that was specified in the SYSPITR conditional restart parameter.

System action: DB2 startup terminates.

Operator response: Collect the SYS1.LOGREC and SVC dump, and contact the system programmer.

System programmer response: Ensure that the conditional restart control record is correct, and then restart DB2. You can run the Print Log Map utility (DSNJU004) to examine the current conditional restart record.

DSNR052I *csect-name* **DB2 MEMBER TO REMAIN ACTIVE IN LIGHT MODE TO RESOLVE INDOUBT UNITS OF RECOVERY**

Explanation: This message is issued at the end of restart when the DB2 member has been restarted with the LIGHT(YES) parameter and there are indoubt units of recovery (URs).

System action: The DB2 member remains up and running in light mode. This DB2 member will only allow connections for the purpose of resolving the indoubt URs. Also, certain commands are allowed. Any attempt to connect to this DB2 to submit new SQL work will be disallowed with reason code 00F300A2.

User response: Ensure that the appropriate commit coordinators are started so that they can resync with the DB2 member to resolve the indoubts. Alternatively the -RECOVER INDOUBT command can be used to manually resolve the indoubts.

DSNR053I *csect-name* **ALL INDOUBT UNITS OF RECOVERY HAVE BEEN RESOLVED. LIGHT MODE DB2 NORMAL TERMINATION TO FOLLOW**

Explanation: This message is issued following DSNR052I after all indoubt units of recovery have been resolved for this DB2 member, which was restarted with the LIGHT(YES) parameter.

System action: DB2 subsystem shutdown is initiated.

DSNR054I **RESTART...BEGIN SCAN FOR CHECKPOINT**

Explanation: DB2 must scan the log backwards to locate the prior checkpoint.

System action: After the prior checkpoint is located on the log, restart continues.

DSNR055I *csect-name* **RESTART SUSPENDED SINCE** *time* **IN** *module offset maint-lvl*

Explanation: This message is produced periodically during the recovery phase of a restart if DB2 detects that the restart processor has remained suspended across a two minute interval. This message provides the time that the restart processor has suspended, and, the module, offset, and maintenance level for the suspend request.

System action: DB2 restart continues.

Operator response: Notify the system programmer.

System programmer response: If the message repeats every two minutes and DB2 restart does not appear to be progressing, look for additional messages that could indicate the reason for the restart stall. If no other messages are found, request SVC dumps of the DB2 and IRLM address spaces for problem determination. For data sharing, request dumps from all members of the data sharing group.

DSNR056I *csect-name* **RESTART SUSPENDED ON IRLM REQUEST. ONE HOLDER OF RESOURCE IS** *ss-name***. SERVICE INFO: IRLM FUNC** *func-cd***, RESOURCE** *res-name***.**

Explanation: This message is produced periodically during the recovery phase of restart if DB2 detects that the restart processor has remained suspended across a two minute interval on an IRLM request. If DB2 is suspended for an IRLM lock in a data sharing environment, DB2 queries IRLM to determine what DB2 subsystem holds the lock. If successful, *ss-name* contains the DB2 subsystem name. Otherwise, *ss-name* contains a series of eight asterisks ('********'). The function code and resource name for the IRLM request is also provided for serviceability.

System action: DB2 restart continues.

Operator response: Notify the system programmer.

System programmer response: If the message repeats every two minutes and DB2 restart does not appear to be progressing, the DB2 subsystem displayed in the message might need to be recycled to allow this subsystem to restart. Request SVC dumps of the DB2 and IRLM address spaces before recycling any of the DB2 subsystems for problem determination. For data sharing, request dumps from all members of the data sharing group.

DSNR057I *csect-name* **RESOURCE INFORMATION: DBID X'***dbid***', PSID X'***psid***', PART X'***part#***'**

Explanation: This message is produced periodically during the recovery phase of restart if DB2 detects that the restart processor has remained suspended across a two minute interval on an IRLM request. If DB2 is suspended for an IRLM lock in a data sharing environment, DB2 queries IRLM to determine what DB2 subsystem holds the lock. This message indicates that the locked resource is a type which is held on a data base object. The *dbid* contains the DBID and the *psid* contains the PSID in the case of a table space or ISOBID in the case of an index space. If the object is a partition of a partitioned pageset, then the *part#* contains the partition number, otherwise it is zero.

System action: DB2 restart continues.

Operator response: Notify the system programmer.

System programmer response: If the message repeats every two minutes and DB2 restart does not appear to be progressing, then an agent running on the DB2 subsystem identified in message DSNR056I could be holding on to a lock for an extended period of time, that is associated with the DBID, PSID, PART identified in this message. You might be able to use the data base object identification information in this message to identify the specific object which is encountering the problem. One way to check what object is encountering the problem is to query the DB2 catalog to find the data base name and pageset name, and then to use the DISPLAY DATABASE CLAIMERS or DISPLAY DATABASE LOCKS commands to identify the agent that is holding the lock. If an agent can be identified, you can cancel the agent to release the lock.

DSNR058I *csect-name* **ACCESSING ARCHIVE LOG** *data-set-name*

Explanation: This message is produced periodically during the recovery phase of restart if DB2 detects that the restart processor has remained suspended across a two minute interval attempting to access an archive log data set. The *data-set-name* contains the data set name of the archive log that DB2 is accessing.

System action: DB2 restart continues.

Operator response: Notify the system programmer.

System programmer response: Look for and resolve any outstanding archive log tape mounts or HSM recall requests that could be causing DB2 restart to stall. If this does not resolve the problem, request SVC dumps of the DB2 and IRLM address spaces for problem determination.

Chapter 16. DSNS... messages

The following information messages indicate the progress of DB2 when specifying storage parameters.

DSNS001I *csect_name* **SUBSYSTEM HAS EXCEEDED THE USER SPECIFIED REAL STORAGE USAGE THRESHOLD**

Explanation: You have requested to perform real storage monitoring by specifying a non-zero value for subsystem parameter SPRMRSMX. DB2 has determined that the amount of real storage used has reached or exceeded the amount of combined real and auxiliary storage allowed.

System action: The DB2 subsystem requests to record a DB2 SYS1.LOGREC entry, and an RC00E20033 abend dump is requested before DB2 subsystem termination is initiated. The system terminates with:

```
*DSNV086E = DB2 ABNORMAL TERMINATION
            REASON=00E20033
```

System programmer response: Determine whether the value specified for subsystem parameter, SPRMRSMX is realistic. If you decide to contact IBM support, provide the DB2 SYS1.LOGREC entry and the RC00E20033 dump. Also, you will most likely be asked to provide IFCID225 records, which show historical storage usage, in order to determine what area is responsible for the real frame build up.

DSNS002I *csect_name* **THE VALUE SPECIFIED FOR SUBSYSTEM PARAMETER, SPRMRSMX, IS EQUAL OR GREATER THAN AVAILABLE REAL STORAGE**

Explanation: The value that you specified for the subsystem parameter, SPRMRSMX, is equal to or greater than the amount of real storage that is available to the address space.

System action: DB2 continues with normal startup.

System programmer response: The value specified might be normal if the installation expects that much of the real storage usage to be paged out to auxiliary storage devices.

© Copyright IBM Corp. 1983, 2008

Chapter 17. DSNT... messages

Messages issued by DSNTINST, DSNTINS1, and DSNTINS2 installation CLISTs have an internal severity level as follows:

0 Successful execution: CLIST progress and tracing messages (processing continues)

4 Warning: some adjustments might be made by the installation CLISTs (processing and JCL editing continues)

8 Error: unable to continue (no JCL is edited)

DSNT101I A VSAM ERROR HAS BEEN ENCOUNTERED WHILE CSECT DSNTSTR2 WAS READING THE DB2 BSDS VSAM RETURN CODE = *x*, VSAM REASON CODE = *y*

Explanation: CSECT DSNTSTR2 attempted to read the DB2 bootstrap data set (BSDS), and a VSAM error has occurred. In the message, 'x' is a 1-byte VSAM return code and 'y' is a 1-byte VSAM reason code.

System action: The start up process continues. However, any references to password-protected system databases will cause a 'resource unavailable' error. Any agent-related requests for subsystem database resources will result in an SQLCODE -904. Any requests that are not agent-related for unavailable resources will result in the following messages:

DSNI001I	DSNI002I	DSNI003I	DSNI004I
DSNI005I	DSNI006I	DSNT408I	DSNT414I
DSNT415I	DSNT416I	DSNT417I	DSNT500I

Abend X'04E' with reason code X'00E30010' is also produced. However, the error results in a retry. DB2 does not terminate, but a dump is produced.

Operator response: Notify the system programmer.

System programmer response: A VSAM error occurred while reading the BSDS record with key '200000001'. Either the BSDS record does not exist or an I/O error occurred (indicating that the record is bad). Further information may be obtained from a VSAM diagnostics manual. Refer to Part 4 (Volume 1) of *DB2 Administration Guide* for procedures on recovering or repairing the DB2 bootstrap data set.

Problem determination: Collect the following diagnostic items listed in Appendix A, “Problem determination,” on page 659: 1, 2, 5, 16.

DSNT102I CSECT DSBTSTR2 ENCOUNTERED AN INVALID DATABASE SERVICES BOOTSTRAP RECORD

Explanation: CSECT DSNTSTR2 attempted to read the DB2 bootstrap data set (BSDS), and an invalid record was read by the recovery log manager (RLM) subcomponent. This is a DB2 error.

System action: The DB2 start up process continues. However, any references to DB2 password-protected system databases will cause a 'resource unavailable' error. Any agent-related requests for DB2 subsystem database resources will result in an SQLCODE -904. Any requests that are not agent-related for unavailable resources will result in the following messages:

DSNI001I	DSNI002I	DSNI003I	DSNI004I
DSNI005I	DSNI006I	DSNT408I	DSNT414I
DSNT415I	DSNT416I	DSNT417I	DSNT500I

Abend X'04E' with reason code X'00E30010' is also produced. However, the error results in a retry. DB2 does not terminate, but a dump is produced.

Operator response: Notify the system programmer.

System programmer response: Obtain a copy of SYS1.LOGREC and the SVC dump. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: This message is generated when the recovery log manager (RLM) subcomponent of DB2 encounters an invalid DB2 database services password record. A copy of the BSDS record with key '200000001', which is the invalid database services record retrieved, should be obtained.

Collect the following diagnostic items listed in Appendix A, “Problem determination,” on page 659: 1, 2, 5, 16.

DSNT103I THE CURRENTLY LOADED DSN6SPRM IS NOT THE MVS/XA VERSION

Explanation: A -START DB2 PARM=xxxxxxxx was issued where 'xxxxxxxx' is the version of the system parameter CSECTs. DB2 executes in a MVS/XA

© Copyright IBM Corp. 1983, 2008

environment and will not execute in any other environment.

System action: The message DSNV086E is generated with a reason code of X'00E30008'. DB2 is terminated with abend code X'04F'.

Operator response: Notify the system programmer.

System programmer response: Restart DB2 with a -START DB2 PARM= command, where the parameter block that follows the 'PARM=' parameter specifies an MVS/XA version of the DB2 subsystem parameters.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 3, 5, 9.

DSNT104I THE CURRENTLY LOADED DSN6SPRM IS NOT THE MVS/370 VERSION

Explanation: A -START DB2 PARM=xxxxxxxx was issued, where 'xxxxxxxx' is the version of the system parameter CSECTs. DB2 executes in an MVS/370 environment and does not execute in any other environment.

System action: Message DSNV086E is generated with reason code X'00E30009'. DB2 is terminated with abend code X'04F'.

Operator response: Notify the system programmer.

System programmer response: Restart DB2 with a -START DB2 PARM= command, where the parameter block that follows the PARM= parameter specifies an MVS/370 version of the DB2 subsystem parameters.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 3, 5, 9.

DSNT105I THE LOADED DSN6SPRM CSECT IS OBSOLETE. PLEASE REASSEMBLE THE DSN6SPRM CSECT AND RELINKEDIT THE DSNZPARM LOAD MODULE.

Explanation: DSNTSTR2 loaded an out-of-date system parameter CSECT DSN6SPRM. This error can only occur if code maintenance has been applied and CSECT DSN6SPRM has not been reassembled. This CSECT is contained in the load module specified in the 'PARM=' parameter of the -START DB2 command.

System action: The message DSNV086E is generated with a reason code of X'00E30008'. DB2 is terminated with abend code X'04F'.

Operator response: Notify the system programmer.

System programmer response: Reassemble DSN6SPRM, and relink-edit the DSNZPARM load module. This is done by resubmitting the install job stream, DSNTIJUZ, that built the load module that is named following the 'PARM=' parameter of the -START DB2 command.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 3, 5, 9.

DSNT106I THE LOADED DSN6SYSP CSECT IS OBSOLETE. PLEASE REASSEMBLE THE DSN6SYSP CSECT AND RELINKEDIT THE DSNZPARM LOAD MODULE.

Explanation: DSNTSTR2 loaded an out-of-date system parameter CSECT DSN6SYSP. This error can only occur if code maintenance has been applied and CSECT DSN6SYSP has not been reassembled. This CSECT is contained in the load module specified in the 'PARM=' parameter of the -START DB2 command.

System action: The message DSNV086E is generated with a reason code of X'00E30008'. DB2 is terminated with abend code X'04F'.

Operator response: Notify the system programmer.

System programmer response: Reassemble DSN6SYSP, and relink-edit the DSNZPARM load module. This is done by resubmitting the install job stream, DSNTIJUZ, that built the load module that is named following the 'PARM=' parameter of the -START DB2 command.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 3, 5, 9.

DSNT107I INVALID AMODE OR RMODE ATTRIBUTE FOUND FOR LOAD MODULE *module-name*

Explanation: During DB2 startup, the service controller establishes the addressability to the DB2 database services functional recovery and the emergency shutdown routines. Two load modules, DSNTF, and DSNTLCLE, are loaded into CSA. In the MVS/XA environment, DSNTLCLE must be link-edited with attributes AMODE(24) and RMODE(ANY). At least one of the modules did not have the correct addressing attribute when it was loaded. 'module-name' in the error message is the name of the first load module that has an invalid addressing mode.

System action: DB2 database services start up is terminated.

Operator response: Notify the system programmer.

System programmer response: Verify that all install and maintenance activities against these modules were executed using SMP. Verify that the JCLIN for SMP includes the correct AMODE and RMODE control statements, and that the linkage editor provided with the Data Facility Product (DFP) was invoked by SMP for the link edits. Correct the procedure or JCL which

caused the error and rerun the necessary SMP jobs.

Problem determination: A DB2 dump was requested to an MVS/XA SYS1.DUMP data set.

The MVS service aid AMBLIST provides linkage editor module attributes in the summary section of the LISTLOAD function output.

DSNT108I DB2 HAS DETECTED A DIFFERENCE BETWEEN THE RECORDED SYSTEM CCSIDS AND THE CCSIDS SPECIFIED BY DSNHDECP

Explanation: During DB2 startup, DB2 detected a difference between the CCSIDs recorded in the BSDS and the CCSIDs specified by the version of DSNHDECP that was loaded during startup. DB2 startup terminated to avoid potential data corruption issues.

System action: DB2 startup is terminated.

Operator response: Notify the system programmer.

System programmer response: Determine the reason for the mismatch between the CCSIDs in the BSDS and the CCSIDs specified by the DSNHDECP module that was loaded during startup. Possible reasons for a mismatch include that the procedure used to start DB2 might have incorrect library concatenations, or, the copy of DSNHDECP that was loaded was incorrectly copied to the libraries specified by the DB2 startup procedure.

Correct the mismatch. If IBM support directs you, you can use the change log inventory (DSNJU003) utility with the DELETE CCSIDS clause to delete the CCSIDs that are specified in the in the BSDS. **Important:** Do not attempt to delete the CCSIDs in the BSDS unless IBM support instructs you.

DSNT109I DB2 HAS DETECTED AN INVALID CCSID IN THE CCSID(S) SPECIFIED FOR ENCODING SCHEME *encoding-scheme* **BY DSNHDECP**

Explanation: The version of DSNHDECP that was loaded during DB2 startup has invalid CCSIDs for the *encoding-scheme* encoding scheme. DB2 startup terminated to avoid potential data corruption issues.

System action: DB2 startup is terminated.

Operator response: Notify the system programmer.

System programmer response: Appendix A of DB2 Installation Guide specifies valid CCSIDs for ASCII, EBCDIC, and Unicode. DB2 sets the *encoding-scheme* keyword to ASCII, EBCDIC, or Unicode to identify the source of the error.

If the DSNHDECP value for MIXED is NO, the specified CCSIDs must be SBCS CCSIDs. You can find these CCSIDs in the table entitled ″Single-byte coded character set identifiers (CCSIDs)″.

If the DSNDECP value for MIXED is YES, then you must select ASCII MIXED from the table entitled ″ASCII double-byte coded character set identifiers (CCSIDs)″, and you must select EBCDIC MIXED CCSIDs from the table entitled ″EBCDIC double-byte coded character set identifiers (CCSIDs)″.

Unicode CCSID values are pre-filled and you should not change them.

DSNT110I *mod-name* **UNEXPECTED ERROR FROM IXLMG SERVICE. IXLMG RETURN CODE =** *rc***. IXLMG REASON CODE =** *rsncode***.**

Explanation: The z/OS IXLMG data collection service issued an error while collecting DB2 statistics. The *rc* and *rsncode* variables indicate the return and reason code. Refer to ″z/OS MVS Sysplex Services Reference″ for a description of the IXLMG return and reason codes.

System action: DB2 inserts a value of x'FFFFFFFF' into the QTGSFLMG field of the DB2 Statistics record.

DSNT200I *type bind-type* **FOR PLAN** *plan-id* **SUCCESSFUL**

Explanation: A BIND, REBIND or FREE PLAN subcommand has completed successfully.

type Indicates that a test bind was in progress. A test bind means that no plan name was specified. The token is omitted if a normal BIND PLAN was processed.

bind-type
Type of bind subcommand: BIND, REBIND, or FREE.

plan-id Name of application plan specified in the subcommand.

System action: The plan was successfully bound, rebound, or freed.

DSNT201I *type bind-type* **FOR PLAN** *plan-id* **NOT SUCCESSFUL**

Explanation: Errors have been detected during the processing of the indicated BIND, REBIND, or FREE PLAN subcommand.

type Indicates that a test bind was in progress. A test bind means that no plan name was specified. The token is omitted if a normal BIND PLAN was processed.

bind-type
Type of bind subcommand: BIND, REBIND, or FREE.

plan-id Name of application plan specified in the subcommand.

System action: The BIND, REBIND, or FREE PLAN was not performed.

System programmer response: See other messages issued to determine the error that caused the bind subcommand to terminate unsuccessfully.

DSNT202I MESSAGE LIMIT EXCEEDED

Explanation: A fixed amount of space exists in which to generate bind subcommand messages. This message says that the bind subcommand generated more messages than will fit in that space.

System action: For bind subcommands that affect a single plan, the bind subcommand process continues. However, no other messages are generated except for message DSNT200I or DSNT201I. For bind subcommands that affect more than one plan, the bind subcommand process terminates and no other messages are generated.

System programmer response: The 'FLAG' parameter of the bind subcommand can be used to limit the types of messages that are generated.

DSNT205I *name* **IS A DUPLICATE** *name-type*, **IT WILL BE USED ONLY ONCE**

Explanation: Name 'name' has been specified more than once in a list of names.

name Name that was specified more than once.

name-type
Indicates the type of list in which the name was duplicated.
- MEMBER NAME - List of DBRM Members
- PLAN-ID - List of plan IDs
- SYSTEM NAME - List of system connection names on a DISABLE or ENABLE list.
- PACKAGE NAME - List of package IDs

System action: Duplicates are removed from 'name-type' list and the bind process continues.

System programmer response: Verify the list that contains the duplicate. If the error is only in entering a name incorrectly, which happens to duplicate another entry, enter the name correctly and reissue the bind statement. If the entry is really a duplicate, no action is required.

DSNT207I PROGRAM *program-name* **WITH MARK** *release-dependency-mark* **FAILED BECAUSE IT DEPENDS ON FUNCTIONS OF THE RELEASE FROM WHICH FALL BACK HAS OCCURRED**

Explanation: Program 'program-name' depends on a function of DB2 that is not supported by the current active release.

program-name
Name of the application program.

release-dependency-mark
A one-character mark showing the oldest DB2 release supporting this program.

System action: The bind operation for this plan or package is not performed.

User response: The program cannot be used until the DB2 subsystem is remigrated to the newer release. Notify the system programmer.

System programmer response: Warn users not to use plans or packages containing this program until the DB2 subsystem has been remigrated to the newer release.

DSNT208I PROGRAM *program-name* **PRECOMPILED WITH INCORRECT LEVEL FOR THIS RELEASE**

Explanation: Program 'program-name' was precompiled under a release not supported by the current level of DB2, or the contents of the DBRM have been modified after the precompilation phase.

System action: The BIND operation was not performed.

User response: Precompile the named program again using the current precompiler. Reissue the BIND command.

Problem determination: If the application program was precompiled at an appropriate release, and the problem persists, collect the following diagnostic items listed in Appendix A, “Problem determination,” on page 659: 79, 80, 81.

DSNT209I *bind-type* **FOR PLAN** *plan-id* **WITH MARK** *release-dependency-mark* **FAILED BECAUSE PLAN DEPENDS ON FUNCTIONS OF THE RELEASE FROM WHICH FALL BACK HAS OCCURRED.**

Explanation: Plan 'plan-id' depends on a function of DB2 which is not supported by the currently active release.

bind-type
REBIND

plan-id Name of the application plan

release-dependency-mark
A one-character mark showing the oldest release of DB2 which can support this plan. The release dependency mark for the plan is kept in the IBMREQD columns in the DB2 catalog in SYSIBM.SYSPLAN.

System action: The REBIND operation for this plan is not performed.

User response: The plan cannot be used until the DB2 subsystem is remigrated to the newer release.

Operator response: Notify the system programmer.

System programmer response: Warn users not to use the plan until the DB2 subsystem has been remigrated to the newer release.

DSNT210I *bind-type* **AUTHORIZATION ERROR USING** *authorization-id* **AUTHORITY PLAN=***plan-id* **PRIVILEGE=***privilege*

Explanation: The indicated authorization ID does not have the indicated privilege and therefore cannot invoke the indicated BIND subcommand against the indicated application plan.

bind-type
Type of BIND subcommand: BIND, REBIND, FREE.

authorization ID
Authorization ID of the plan owner.

plan-id Application plan specified in the BIND subcommand.

privilege
Missing privilege: BIND, BINDADD

System action: The BIND, REBIND, or FREE operation for 'plan-id' is not performed.

System programmer response: If the indicated privilege is BINDADD, then the privilege to invoke the BIND subcommand with the ACTION(ADD) option must be granted to the indicated authorization ID. If the indicated privilege is BIND, the privilege to invoke a BIND subcommand against the indicated application plan must be granted to the indicated authorization ID.

DSNT211I **REBIND FOR PLAN** *plan-id* **FAILED BECAUSE IBMREQD OF** *ibmreqd* **IS INVALID**

Explanation: The IBMREQD column of the SYSIBM.SYSPLAN catalog table for the named plan contains an unrecognizable character.

System action: The REBIND operation for the plan was not performed.

System programmer response: You must do a BIND ACTION(REPLACE) for this plan.

DSNT212I *bind-type* **AUTHORIZATION ID** *auth-id* **IN THE OWNER KEYWORD IS INVALID**

Explanation: The authorization ID specified in the OWNER keyword of the BIND or REBIND subcommand is invalid. To specify an OWNER that is not the primary authorization ID, or one of the secondary IDs of the process requires additional authority. See *DB2 Command Reference* for the authority required.

bind-type
Type of bind subcommand: BIND or REBIND

auth-id The authorization ID specified as owner.

System action: The BIND or REBIND subcommand is not executed.

System programmer response: Rerun the subcommand and specify a valid authorization ID for the OWNER keyword. You may need to add the required authorization ID for this user to the security system.

DSNT213I **THE DCB BLOCK SIZE OF THE INPUT DBRM LIBRARY IS INVALID**

Explanation: An invalid DCB block size is specified for the DBRM library of the BIND subcommand.

System programmer response: Correct the DCB block size of the DBRM library, and reissue the BIND subcommand.

DSNT214I **I/O ERROR ON DBRM LIBRARY DBRM MEMBER=** *dbrm-name* **DBRM LIBRARY=** *dbrm-pds-name xxxxxx*

Explanation: An I/O error was encountered attempting to read database request module (DBRM) during a BIND subcommand. 'xxxxxx' is a portion of the message text generated by the SYNADAF macro instruction to describe the permanent I/O error This text may contain information such as jobname, stepname, unit address, device type, ddname, operation attempted, and error description.

dbrm-name
Name of the DBRM associated with the error.

dbrm-pds-name
Partitioned data set associated with the error.

System action: No application plan is created.

System programmer response: Determine if an incorrect dbrm-pds-name was specified in the LIBRARY parameter of the BIND subcommand, or an incorrect data set name was specified for the DBRMLIB DD statement. Check correctly specified data sets for I/O errors or modified DCB attributes.

DSNT220I **BIND ADD ERROR USING** *auth-id* **AUTHORITY PLAN** *plan-id* **ALREADY EXISTS**

Explanation: An attempt was made to use the BIND subcommand with the 'ACTION(ADD)' option that specified a 'plan-id' that already exists.

auth-id Authorization ID of the owner of the plan.

plan-id Application plan specified in the BIND subcommand.

System action: No application plan is created.

System programmer response: Check the SYSIBM.SYSPLAN catalog table for names of existing application plans. Reinvoke the BIND subcommand with a 'plan-id' that is not in use.

DSNT221I *bind-type* **ERROR USING** *auth-id* **AUTHORITY PLAN** *plan-id* **DOES NOT EXIST**

Explanation: The indicated bind subcommand was issued against an application plan that does not exist. The individual variable fields contain:

bind-type
Type of bind subcommand: REBIND or FREE.

auth-id Authorization ID of the plan owner.

plan-id Application plan specified in the BIND subcommand.

System programmer response: Check the SYSPLAN catalog table for the correct 'plan-id' to use.

DSNT222I *csect-name bind-type* **WARNING FOR** *object-type = object-name* **USE OF OPTHINT RESULTS IN** *count-1* **STATEMENTS WHERE OPTHINT FULLY APPLIED** *count-2* **STATEMENTS WHERE OPTHINT NOT APPLIED OR PARTIALLY APPLIED** *count-3***STATEMENTS WHERE OPTHINT IS NOT FOUND**

Explanation: The DB2 subsystem parameters have not been configured to enable the use of OPTHINT in the *bind-type* subcommand. The default OPTHINT value of all blanks is used.

csect-name
Name of the csect from which the message was issued.

bind-type
Type of BIND subcommand: BIND or REBIND.

object-type
PLAN or PACKAGE

object-name
If *object-type* is PLAN, *object-name* is the name of the plan.

If *object-type* is PACKAGE, *object-name* is the name of the package in the format 'location-id.collection-id.package-id'.

count 1 The number of statements where hints were found, and all hints found were successfully applied.

count 2 The number of statements where hints were found and at least one of those hints could not be successfully applied.

count 3 The number of statements where no hint was found.

System action: Processing continues normally.

| **Programmer response:** Examine the counts to ensure that the specified optimization hints were applied to the intended statements. Refer to SQL codes +394 and +395 for more information about individual statements.

User response: Determine if the OPTHINT usage reported in the message matches what is expected. If not, refer to *DB2 Administration Guide* for more information on OPTHINT usage.

DSNT223I *csect-name bind-type* **WARNING FOR** *object-type = object-name* **USE OF OPTHINT IS DISALLOWED BY A DB2 SUBSYSTEM PARAMETER. THE DEFAULT OPTHINT VALUE IS USED.**

Explanation: The DB2 subsystem parameters have not been configured to enable the use of OPTHINT in the *bind-type* subcommand. The default OPTHINT value of all blanks is used.

bind-type
Type of BIND subcommand: BIND or REBIND.

object-type
PLAN or PACKAGE

object-name
If *object-type* is PLAN, *object-name* is the name of the plan.

If *object-type* is PACKAGE, *object-name* is the name of the package in the format 'location-id.collection-id.package-id'.

System action: Processing continues normally using the default OPTHINT value.

System programmer response: Enable the use of OPTHINT by changing the value of OPTIMIZATION HINTS on the DB2 Installation panel, DSNTIP4.

If, after further consideration, you do not want to use OPTHINT, remove OPTHINT from the *bind-type* subcommand or set the value to all blanks.

DSNT224I *csect-name bind-type* **ERROR FOR** *object-type object-name* **CCSID =** *ccsid* **IS INVALID FOR ENCODING SCHEME =** *scheme*

Explanation: The system does not have a valid CCSID

defined for the specified encoding scheme, or the specified CCSID is not defined on the system.

bind-type
Type of bind subcommand: BIND or REBIND.

object-type
PLAN or PACKAGE.

object-name
If object-type is PLAN, object-name is the name of the application. Otherwise, it is the name of the package in 'location.collection.package' format.

scheme The encoding scheme specified in the ENCODING bind option on the subcommand or defaulted from the system default Application Encoding Scheme. If *scheme* is UNKNOWN, then the CCSID value specified in the ENCODING bind option or defaulted from the system default Application Encoding Scheme is not defined, and therefore the encoding scheme is not known.

ccsid The numeric CCSID corresponding to the specified *ccsid-type*.

Severity: 8 (error)

System action: The *bind-type* operation is not performed

User response: Verify that the correct encoding scheme was specified in the ENCODING bind option on the *bind-type* subcommand. If correct, see System Programmer Response and contact the System Programmer. If incorrect, specify the correct scheme in the ENCODING option and reissue the *bind-type* subcommand.

System programmer response: If the bind option ENCODING was not specified, and instead the value for ENCODING was defaulted from the system default Application Encoding Scheme, verify the Application Encoding Scheme is set to a defined CCSID.

If the value specified for the ENCODING option was ASCII, EBCDIC or UNICODE, verify the corresponding CODED CHAR SET subsystem parameter on the installation panel DSNTIPF has been set to a defined CCSID.

Otherwise, inform the user of the correct CCSID. Refer to the section entitled "Character conversion" of *DB2 Installation Guide* for more information regarding CCSIDs and character sets.

DSNT228I BIND ERROR, ATTEMPTING TO REPLACE PACKAGE = *package_name* **WITH VERSION =** *(version2)* **BUT THIS VERSION ALREADY EXISTS**

Explanation: An attempt is made to create a version of a package that already exists. The version specified in the REPLVER keyword is different from the version specified for the precompile. The version specified for the precompile already exists in the catalog. The combination of 'location.collection.package.version' must be unique in the SYSIBM.SYSPACKAGE catalog table. A common mistake is that the user may believe that the version he is creating is the one specified in the REPLVER keyword. This is not the case. The version specified in the REPLVER keyword is the name of the version being replaced. The version that will be created is the version that was given to the program when it was precompiled.

package_name
Fully qualified package name

version2
Version-id of package that is to be created

System action: The bind will fail.

System programmer response: There are two approaches to solve this problem. The first is to re-precompile the program with a new version name and re-issue the original BIND subcommand. The other approach is not to do the precompile but re-issue the BIND subcommand without the REPLVER parameter.

DSNT229I BIND ERROR, ATTEMPTING TO REPLACE PACKAGE = *package-name* **THERE ARE ENABLE OR DISABLE ENTRIES CURRENTLY ASSOCIATED WITH THE PACKAGE**

Explanation: The BIND subcommand was issued to replace a package that has ENABLE or DISABLE entries currently associated with the package. The individual variable fields contain:

package-name
Name of the package (location.collection.package)

System action: Package not bound.

System programmer response: FREE the package first and then BIND the package.

DSNT230I BIND DBRM MEMBER NAME ERROR USING *auth-id* **AUTHORITY** *object-type* = *object-name* **MEMBER** *dbrm-name* **NOT FOUND IN PDS SEARCH ORDER**

Explanation: The indicated 'dbrm-name' was not found in the search order in the PDSs specified on the BIND PLAN or BIND PACKAGE subcommand.

auth-id Authorization ID of the plan or package owner

object-type
The type of object being bound: PLAN or PACKAGE

object-name
If 'object-type' is PLAN, 'object-name' is the

name of the application plan. Otherwise, it is the name of the package in 'location.collection.package' format.

dbrm-name
Member name of the DBRM that was not found.

System action: The bind is unsuccessful.

System programmer response: Determine if an incorrect 'dbrm-member-name' or PDS was specified. Reinvoke the BIND subcommand specifying the correct names.

DSNT231I BIND DBRM ERROR USING *auth-id* **AUTHORITY** *object-type* = *object-name* **DBRM** *dbrm-name* **IS IN AN INCONSISTENT STATE**

Explanation: The contents of a DBRM have been modified after the precompilation process, or the DBRM contains incorrect information in the header.

auth-id Authorization ID of the invoker of the BIND subcommand.

object-type
The type of object being bound: PLAN or PACKAGE

object-name
If 'object-type' is PLAN, 'object-name' is the name of the application plan. Otherwise, it is the name of the package in 'location.collection.package' format.

dbrm-name
Name of the DBRM that is in error.

System action: The bind is unsuccessful.

System programmer response: Precompile the application program associated with the indicated DBRM. Reinvoke the BIND subcommand.

DSNT232I SUCCESSFUL *bind-type* **FOR PACKAGE =** *package-name*

Explanation: A BIND, REBIND or FREE PACKAGE subcommand has completed successfully.

bind-type
Type of bind subcommand: BIND, REBIND, or FREE

package-name
Name of the package in the following format: 'location.collection.package.(version)'

System action: The package was successfully bound, rebound, or freed.

DSNT233I UNSUCCESSFUL *bind-type* **FOR PACKAGE =** *package-name*

Explanation: Errors have been detected during the processing of the indicated BIND, REBIND, or FREE PACKAGE subcommand.

bind-type
Type of bind subcommand: BIND, REBIND, or FREE

package-name
Name of the package in the following format: 'location.collection.package.(version)'

System action: The indicated package was not bound, rebound, or freed.

System programmer response: See other messages issued to determine the error that caused the subcommand to terminate unsuccessfully.

DSNT234I MESSAGE LIMIT EXCEEDED

Explanation: A fixed amount of space exists in which to generate bind subcommand messages. This message says that the bind subcommand generated more messages than will fit in that space.

System action: For bind subcommands that affect a single package, the bind subcommand process continues. However, no other messages are generated except for completion message DSNT232I or DSNT233I. For bind subcommands that affect more than one package, the bind subcommand process terminates and no other messages are generated.

System programmer response: The 'FLAG' parameter of the bind subcommand may be used to limit the types of messages that are generated.

DSNT235I *csect-name bind-type* **AUTHORIZATION ERROR USING** *auth-id* **AUTHORITY PACKAGE =** *package-name* **PRIVILEGE =** *privilege*

Explanation: The authorization ID given does not have the *privilege* indicated and cannot invoke the bind subcommand on the named package.

bind-type
Type of bind subcommand: BIND, REBIND, or FREE

auth-id Authorization ID of of the package owner

package-name
Name of the package in the following format: 'location.collection.package.(version)'

privilege
Name of the privilege not held:

BINDADD or BINDADD *
The authority to create a new package using BIND with the ADD option.

BIND The authority to BIND (REPLACE) or REBIND a package.

BIND *
The authority to add a new version of an existing package using BIND with the ADD option.

FREE There is no FREE privilege, but the user needs the proper authority to free the package.

COPY The authority to COPY from the indicated package.

CREATE IN or CREATE IN *
The authority to create a package in the indicated collection.

A blank and asterisk (*) in the privilege values 'BINDADD *', 'BIND *', and 'CREATE IN *' in the message indicates that the DB2 system parameter BIND NEW PACKAGE has the value BIND. When ' *' does not appear in these privilege values, BIND NEW PACKAGE has the value BINDADD.

The system parameter BIND NEW PACKAGE affects the authorization required to add a new package or a new version of an existing package to a collection. For more information about this system parameter, see:

- "Installing, Migrating, and Updating System Parameters" (Panel: DSNTIPP) in Part 2 of *DB2 Installation Guide*.
- "Controlling Access to DB2 Objects" in Part 3 (Volume 1) of *DB2 Administration Guide* .

System action: The indicated package is not bound, rebound, or freed.

System programmer response: You must grant the indicated privilege to the authorization ID of the package owner.

DSNT236I **REBIND FOR PACKAGE** *package-name* **FAILED BECAUSE IBMREQD OF** *ibmreqd* **IS INVALID**

Explanation: The IBMREQD column of the SYSIBM.SYSPACKAGE catalog table for the named package contains an unrecognizable character. This usually means that the package depends on a function of DB2 which is not supported by the currently active release.

This message is issued by the following CSECT: DSNTBRB2

package-name
Name of the package in the following format: 'location.collection.package.(version)'

System action: The package was not rebound.

System programmer response: Warn users not to use the package until the DB2 subsystem is remigrated to the newer release.

DSNT237I *csect-name* **BIND** *sub-type* **ERROR USING** *auth-id* **AUTHORITY PACKAGE =** *package-name* **ALREADY EXISTS**

Explanation: An attempt is made to add a package that already exists. The combination of 'location.collection.package.version' must be unique in the SYSIBM.SYSPACKAGE table. In addition, the combination of 'location.collection.package.consistency-token' must be unique.

sub-type
The sub-type can have the values: COPY, ADD or REPLACE.

auth-id Authorization ID of the owner of the package.

package-name
Name of the package in the following format: 'location.collection.package.(version)'

System action: No package is created.

System programmer response: Check the SYSIBM.SYSPACKAGE catalog table for names of existing application packages. Re-invoke the BIND subcommand with a 'location.collection.package.version' that is not in use.

DSNT238I *bind-type error-type* **USING** *auth-id* **AUTHORITY PACKAGE =** *package-name* **DOES NOT EXIST**

Explanation: The indicated subcommand was issued against a package that does not exist. The individual variable fields contain:

bind-type
Type of bind subcommand (BIND COPY | BIND REPLACE | REBIND | FREE| BIND PLAN | REBIND PLAN | DROP PACKAGE). If BIND or REBIND PLAN is specified, the package-name refers to a package on the PKLIST for the plan.

error-type
Type of error: WARNING or ERROR.

auth-id Authorization ID of the invoker of the BIND subcommand.

package-name
Name of the package in the following format 'location.collection.package.(version)'.

System action: Package not rebound or freed.

System programmer response: Check the SYSPACKAGE catalog table for the correct 'location.collection.package.version' to use.

DSNT239I **PLAN** *plan-id* **CONTAINS NO DBRMS AND NO PACKAGES**

Explanation: A plan with no DBRMs was rebound with the NOPKLIST option specified. This creates an empty plan. No SQL can be executed. The individual variable fields contain:

plan-id Plan name specified on the REBIND command.

System action: This is a warning message. The REBIND process will continue.

System programmer response: If an empty plan is not desired, reissue the REBIND command with the PKLIST keyword specifying a new list.

DSNT240I *csect-name bind-type* **ERROR, ASTERISK NOT ALLOWED ON REMOTE** *bind-type* **PACKAGE**

Explanation: When remotely rebinding or freeing a package, an asterisk (*) cannot be specified anywhere in the name of a package. In other words, it is not possible to remotely rebind or free multiple packages.

System programmer response: Remove the asterisk and rebind, or free a specific package. Refer to *DB2 Command Reference* for the correct syntax for REBIND or FREE PACKAGE.

DSNT241I *bind-type* **AUTHORIZATION** *error-type* **ON PLAN** *plan-id* **AUTHORIZATION ID** *auth-id* **NOT AUTHORIZED TO EXECUTE PACKAGE** *package-name*

Explanation: The indicated authorization ID does not have EXECUTE authority on the specified package in the package list.

bind-type
Type of bind subcommand: BIND or REBIND

error-type
Type of error: WARNING or ERROR. If VALIDATE(BIND) was specified, this is an ERROR message. Otherwise, it is a WARNING message.

plan-id Name of application plan specified in the subcommand.

auth-id Authorization ID of the plan owner.

package-name
Name of the package (collection.package). The location of the package is the local DB2 location name.

System action: If the bind subcommand was invoked with VALIDATE(BIND), the BIND or REBIND operation is not performed. If VALIDATE(RUN) was used then the BIND or REBIND operation will continue.

System programmer response: The indicated plan owner should be granted the EXECUTE privilege on the package.

DSNT242I *bind-type* **ERROR USING** *auth-id* **AUTHORITY NO BIND OR REBIND SUBCOMMANDS ARE ALLOWED FOR THIS AUTHID**

Explanation: The authorization ID is not allowed to perform the indicated 'bind-type' against a plan or package because entries in the RLST table prohibit binding and rebinding by this authorization ID, or all authorization IDs.

bind-type
Type of bind subcommand: BIND or REBIND

auth-id Authorization ID of the plan or package owner.

System action: The plan or package is not bound or rebound.

System programmer response: If the indicated authorization ID should be allowed to bind, change the entry in the active RLST table.

DSNT243I **BIND ERROR FOR PACKAGE =** *pkg-id* **CONTOKEN =** *'contoken'***X IS NOT UNIQUE SO IT CANNOT BE CREATED**

Explanation: An attempt is made to add or replace a package with a consistency token that is not unique for that package. In other words, the combination of 'location.collection.package.consistency-token' already exists.

pkg-id Fully qualified name of the package.

contoken
Consistency token in hexadecimal.

System action: The BIND will fail.

System programmer response: Check the SYSIBM.SYSPACKAGE catalog table for names of existing application packages with the indicated consistency token. Re-issue the BIND subcommand such that the 'location.collection.package.consistency-token' is unique within the catalog. The following SQL statement can be used to query the catalog,

```
SELECT COLLID,NAME
FROM loc-id.SYSIBM.SYSPACKAGE
WHERE HEX(CONTOKEN) = 'contokn';
```

DSNT244I *bind-type* **ERROR** *keyword* **KEYWORD IS NOT SUPPORTED FOR REMOTE PROCESSING**

Explanation: The indicated keyword is not supported for the remote bind or rebind subcommand.

bind-type
Type of bind subcommand: BIND or REBIND

keyword
Name of the keyword that is not supported.

Programmer response: Check *DB2 Command Reference* for the correct syntax.

DSNT245I **LOCATION** *location* **CANNOT BE SPECIFIED. THE LOCAL DB2 IS NOT DEFINED**

Explanation: Since the local DB2 location name is undefined, no location name may be given when specifying a package name on a bind subcommand.

location
Location specified on the bind subcommand.

System action: The bind subcommand was not performed.

System programmer response: The location name of the local DB2 must be defined in BSDS if the location is used in the BIND, REBIND, or FREE subcommand.

DSNT246I *bind-type* **FOR PACKAGE** *package-name* **WITH MARK** *release-dependency-mark* **FAILED BECAUSE PACKAGE DEPENDS ON FUNCTIONS OF THE RELEASE FROM WHICH FALL BACK HAS OCCURRED**

Explanation: The named package depends on a function of DB2 which is not supported by the currently active release.

bind-type
Type of bind subcommand: REBIND

package-name
Name of the package in the following format: 'location.collection.package.(version)'

release-dependency-mark
A one-character mark showing the oldest release of the DB2 which can support this package. The release dependency mark for the package is kept in the IBMREQD column in the DB2 catalog in SYSIBM.SYSPACKAGE.

System action: The 'bind-type' operation for this package is not performed.

User response: The package cannot be bound until the DB2 subsystem is remigrated to the newer release.

System programmer response: Warn users not to use plans containing this package until the DB2 subsystem has been remigrated to the newer release.

DSNT247I *bind-type* **ERROR** *object-type* **NOT FOUND FOR** *object-type* = *object-name*

Explanation: You attempted to FREE or REBIND multiple plans or packages using the asterisk (*) but no plans or packages were found.

bind-type
Type of bind subcommand: REBIND or FREE

object-type
Type of object to be freed or rebound: PLAN or PACKAGE

object-name
If object-type is PLAN, object-name is the name of the application plan. Otherwise, it is the name of the package in 'location.collection.package' format.

System action: The REBIND or FREE operation is not performed.

User response: Check the plan or package name. If it is correct, no plans or packages were found and no operation can be performed; otherwise, correct the plan or package name and reissue the REBIND or FREE subcommand.

DSNT248I *csect-name bind-type* **ERROR FOR** *object-type object-name* **CCSID =** *ccsid* **IS INVALID FOR ENCODING SCHEME =** *scheme*

Explanation: The system does not have a valid CCSID defined for the specified encoding scheme, or the specified CCSID is not defined on the system.

bind-type
Type of bind subcommand: BIND or REBIND.

object-type
PLAN or PACKAGE

object-name
If *object-type* is PLAN, *object-name* is the name of the application. Otherwise, it is the name of the PACKAGE in 'location.collection.package' format.

scheme This field displays the encoding scheme specified in the ENCODING bind option on the *bind-type* subcommand or defaulted from the system default Application Encoding Scheme. If *scheme* is UNKNOWN, then the CCSID value has not been defined in the ENCODING bind option or by the system default Application Encoding Scheme, and therefore the encoding scheme is not known.

ccsid This field displays the numeric CCSID specified in the ENCODING bind option or the numeric CCSID corresponding to the scheme specified in the ENCODING bind

option. Otherwise, *ccsid* displays the numeric CCSID corresponding to the specified *ccsid-type*.

Severity: 8 (error)

System action: The *bind-type* operation is not performed

User response: Verify that the correct encoding scheme was specified in the ENCODING bind option on the *bind-type* subcommand. If correct, see System Programmer Response and contact the system programmer. If incorrect, specify the correct scheme in the ENCODING option and reissue the *bind-type* subcommand.

System programmer response: If the bind option ENCODING was not specified, and the value for ENCODING was defaulted from the system default Application Encoding Scheme instead, verify the Application Encoding Scheme is set to a defined CCSID.

If the value specified for the ENCODING option was ASCII, EBCDIC or UNICODE, verify the corresponding CODED CHAR SET subsystem parameter on the installation panel DSNTIPF has been set to a defined CCSID.

Otherwise, inform the user of the correct CCSID. Refer to the section entitled "Character conversion" of *DB2 Installation Guide* for more information regarding CCSIDs and character sets.

DSNT249I *csect-name bind-type* **WARNING FOR PLAN** *plan-name keyword* **LOCATION-ID** *location-id* **NOT FOUND IN SYSIBM.LOCATION**

Explanation: A 'location-id' was specified on the PKLIST or CURRENTSERVER keyword on a bind subcommand. That location is not defined in the SYSIBM.LOCATIONS table.

bind-type
Type of bind subcommand: BIND or REBIND

plan-name
Name of the plan

keyword
The bind subcommand containing the nonexistent location: PKLIST or CURRENTSERVER

location-id
The nonexistent location

System action: The bind process will continue.

System programmer response: Define the appropriate 'location-ids' in the SYSIBM.LOCATION catalog table prior to executing the plan.

DSNT250I **BIND UNABLE TO OPEN DBRM LIBRARY USING** *authorization-id* **AUTHORITY** *object-type = object-name*

Explanation: The BIND subcommand was unable to open the DBRM partitioned data set(s).

authorization-id
Authorization ID of the invoker of the BIND subcommand.

object-type
PLAN | PACKAGE

object-name
If 'object-type' is PLAN, 'object-name' is the name of the application plan. Otherwise, it is the name of the package in 'location.collection.package' format.

User response: Contact the system programmer, providing all material associated with the error.

System programmer response: Determine why the BIND subcommand was unable to open the DBRM library, and correct the problem.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 5, 32.

DSNT251I **BIND UNABLE TO LOCATE DBRM LIBRARY MEMBERS USING** *authorization-id* **AUTHORITY** *object-type = object-name*

Explanation: The BIND subcommand received an I/O error while attempting to locate the specified DBRM library member(s).

authorization-id
Authorization ID of the invoker of the BIND subcommand.

object-type
PLAN | PACKAGE

object-name
If object-type is PLAN, object-name is the name of the application plan. Otherwise, it is the name of the package in 'location.collection.package' format.

System programmer response: Correct the reason for the I/O error, and reissue the BIND subcommand.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 5, 32.

DSNT252I *csect-name bind-type* **OPTIONS FOR PLAN** *plan-name*

ACTION	*bind-option1*	*bind-option2*
OWNER	*bind-option3*	

VALIDATE	*bind-option4*
ISOLATION	*bind-option5*
ACQUIRE	*bind-option6*
RELEASE	*bind-option7*
EXPLAIN	*bind-option8*
DYNAMICRULES	*bind-option9*

Explanation: This message shows the BIND or REBIND options that were used for the plan specified by *plan-name* during bind or rebind processing.

Severity: 0 (informational)

DSNT253I *csect-name bind-type* **OPTIONS FOR PLAN** *plan-name*

(DEFER | NODEFER)	PREPARE
CACHESIZE	*bind-option1*
QUALIFIER	*bind-option2*
CURRENTSERVER	*bind-option3*
CURRENTDATA	*bind-option4*
DEGREE	*bind-option5*
SQLRULES	*bind-option6*
DISCONNECT	*bind-option7*
REOPT	*bind-option8*
KEEPDYNAMIC	*bind-option9*
IMMEDWRITE	*bind-option10*
DBPROTOCOL	*bind-option11*
OPTHINT	*bind-option12*
ENCODING	*bind-option13* (*bind-option14*)
PATH	*bind-option15*

Explanation: This message shows the BIND or REBIND options that were used for the plan specified by *plan-name* during bind or rebind processing.

Severity: 0 (informational)

DSNT254I *csect-name bind-type* **OPTIONS FOR**

PACKAGE	=*package-name*
ACTION	(ADD | REP)
OWNER	*owner*
QUALIFIER	*qualifier*
VALIDATE	(BIND | RUN)
EXPLAIN	(NO | YES)
ISOLATION	(RR | CS | RS | UP | blank)
RELEASE	(COMMIT | DEALLOCATE | blank)
COPY	*package-name*

Explanation: This message shows the BIND or REBIND options that were used for the local package specified by *package-name* during bind or rebind processing.

Severity: 0 (informational)

DSNT255I *csect-name bind-type* **OPTIONS FOR**

SQLERROR	*bind-option1*
CURRENTDATA	*bind-option2*
DEGREE	*bind-option3*
DYNAMICRULES	*bind-option4*
(DEFER | NODEFER)	PREPARE
REOPT	*bind-option5*
KEEPDYNAMIC	*bind-option6*
IMMEDWRITE	*bind-option7*
DBPROTOCOL	*bind-option8*
OPTHINT	*bind-option9*
ENCODING	*bind-option10* (*bind-option11*)
PATH	*bind-option12*

Explanation: This message shows the BIND or REBIND options that were used for the local package specified by *package-name* during bind or rebind processing.

Severity: 0 (informational)

DSNT256I **DBRM MEMBER** *member-name* **IS EMPTY, A PACKAGE CANNOT BE BOUND**

Explanation: The named DBRM member is empty, a package cannot be created.

System action: No package is created.

DSNT257I **BIND ERROR, PACKAGE** *package-name* **CANNOT BE CREATED BECAUSE IT HAS THE SAME NAME AS THE PACKAGE BEING COPIED**

Explanation: The location and collection-id of the package being created are the same as the local location and the collection-id specified with the COPY keyword. You cannot create a package that is the same name as the package being copied.

package-name
Name of the package (location.collection.package.(version))

System action: The package is not bound.

System programmer response: Choose a different collection id for the package you are creating and resubmit the subcommand.

DSNT258I *bind-type* **WARNING FOR PLAN** *plan-name* **THE LOCATION-ID** *location-id* **SPECIFIED IN CURRENTSERVER MAY NEED A PACKAGE LIST**

Explanation: If binding a PLAN with a nonlocal

location specified in the CURRENTSERVER, a package list is required using the PKLIST keyword to access any packages at the location specified.

bind-type
Type of bind subcommand: BIND or REBIND

plan-name
Name of plan

location-id
Location of CURRENTSERVER

System action: The bind process will continue.

System programmer response: Add a package list with package(s) that reference the location specified in the CURRENTSERVER or delete the CURRENTSERVER from the BIND or REBIND subcommand.

DSNT259I *bind-type* **WARNING FOR PLAN** *plan-name* **NO PACKAGES IN THE PACKAGE LIST REFERENCE THE LOCATION-ID** *location-id* **SPECIFIED IN CURRENTSERVER**

Explanation: There are no packages in the package list that reference the location specified in the CURRENTSERVER keyword.

bind-type
Type of bind subcommand: BIND or REBIND

plan-name
Name of plan

location-id
Location of CURRENTSERVER

System action: The bind process will continue.

System programmer response: Change 'location-ids' for the packages in the package list to reference the location in the CURRENTSERVER or change the location specified in CURRENTSERVER.

DSNT260I *csect-name bind-type* **- THE SUBMITTED PARAMETERS RESULT IN AN INVALID COMBINATION OF** *bind-option1* **AND** *bind-option2* **FOR** *object-type = object-name*

Explanation: The BIND or REBIND subcommand has been submitted with an invalid combination of bind option or parameter values.

bind-type
Type of bind subcommand: BIND or REBIND.

object-type
PLAN or PACKAGE

object_name
If *object-type* is PLAN, *object-name* is the name of the application plan. If *object_type* is PACKAGE, *object_name* is the name of the package in the format 'location-id.collection-id.package-id'.

bind-option1 **and** *bind-option2*
The invalid combination of bind option values, where *bind-option1* and *bind-option2* are each an option name and value.

If *object-type* is PLAN, then one of the following applies:

- *bind-option1* was used on the REBIND subcommand and *bind-option2* exists in the SYSIBM.SYSPLAN catalog table for PLAN *object-name*, or
- *bind-option2* was used on the REBIND subcommand and *bind-option1* exists in the SYSIBM.SYSPLAN catalog table for PLAN *object-name*.

If *object-type* is PACKAGE, then one of the following applies:

- *bind-option1* was used on the BIND COPY or REBIND subcommand and *bind-option2* exists in the SYSIBM.SYSPACKAGE catalog table for PACKAGE *object-name*, or
- *bind-option2* was used on the BIND COPY or REBIND subcommand and *bind-option1* exists in the SYSIBM.SYSPACKAGE catalog table for PACKAGE *object-name*.

The combination of *bind-option1* and *bind-option2* is not supported by DB2.

System action: The BIND or REBIND process is terminated for the indicated plan or package.

User response: Refer to *DB2 Command Reference* for valid combinations of the indicated bind option values, and reissue the BIND or REBIND subcommand with a valid combination.

Problem determination: See the Explanation section of this message.

DSNT261I *bind-type* **WARNING FOR** *object-type = object-name* **ALL CONNECTION TYPES ARE DISABLED**

Explanation: All possible connection types are disabled. The plan or package cannot be executed.

bind-type
Type of bind subcommand: BIND or REBIND

object-type
PLAN or PACKAGE

object-name
If object-type is PLAN, object-name is the name of the application plan. Otherwise, it is the name of the package in 'location.collection.package' format.

System action: The bind process will continue.

System programmer response: Delete one or more connection types from DISABLE clause or add a list of connection names associated with one of the connection types (DLIBATCH, CICS, IMSBMP, IMSMPP, or REMOTE. REMOTE is only valid for BIND/REBIND PACKAGE)

DSNT262I *csect-name bind-type* **WARNING - ISOLATION** *old-iso* **NOT SUPPORTED. CHANGING TO ISOLATION** *new-iso*

Explanation: DB2 does not support the isolation level specified on the local BIND or REBIND PACKAGE subcommand. The isolation level specified by *new-iso* is used instead.

Severity: 4

System action: Processing continues

User response: If the new isolation level is not satisfactory, change the ISOLATION option on the BIND or REBIND PACKAGE subcommand and rebind the package. Use a value that DB2 supports.

DSNT263I *csect-name bind-type* **FOR** *object-type* = *object-name* **FAILED BECAUSE THE** *bind-type* **DEPENDS ON FUNCTIONS IN A LATER DB2 RELEASE WITH MARK** *release-dependency-mark* **OR HIGHER**

Explanation: The current DB2 does not support a bind option or feature specified in the indicated bind subcommand. The bind option or feature depends on functions in a later DB2 release.

bind-type
Type of bind subcommand: BIND or REBIND

object-type
PLAN or PACKAGE

object-name
If *object-type* is PLAN, *object-name* is the name of the plan.

If *object-type* is PACKAGE, *object-name* is the name of the package in the format `location-id.collection-id.package-id`.

release-dependency-mark
A one-character mark indicating the oldest DB2 release that can support the bind subcommand.

This message can be issued in one of the following situations:

- When the DSN command explicitly or implicitly uses the name of a DB2 subsystem that does not support a bind option or feature used in the DSN bind subcommand *bind-type*.
- When the DSN command uses a group attachment name for a DB2 data sharing group and the receiving DB2 data sharing member does not support a bind option or feature used in the DSN bind subcommand *bind-type*.
- When attempting to use new function before new function mode has been enabled.

System action: The *bind-type* operation is not performed.

User response: In the DSN command, specify the name of a DB2 subsystem having release dependency mark *release-dependency-mark* or higher that supports all of the bind options or features used in the *bind-type* subcommand.

For a list of the DB2 release dependency marks and their meanings, refer to the IBMREQD column in the SYSIBM.SYSPLAN or SYSIBM.SYSPACKAGE DB2 catalog tables described in Appendix F of *DB2 SQL Reference*.

System programmer response: See the 'User Response' section of this message.

DSNT264I *csect-name bind-type bind-subtype* **ERROR USING** *auth-id* **AUTHORITY PACKAGE** *package-name* **IS A TRIGGER PACKAGE**

Explanation: To rebind a trigger package, you must use the DSN subcommand REBIND TRIGGER PACKAGE. To free a trigger package, you must use the SQL statement DROP TRIGGER. You cannot copy a trigger package.

bind-type
Type of bind subcommand: BIND, REBIND, or FREE

bind-subtype
Subtype of bind subcommand: COPY or blank

auth-id Authorization ID attempting the prohibited action

package-name
Name of the package in the following format: 'collection.package'

System action: The trigger package is not copied, rebound, or freed.

System programmer response: Check the SYSIBM.SYSPACKAGE catalog table for names of existing trigger packages. Re-invoke the bind subcommand with a 'collection.package' that is not a trigger package.

DSNT265I *csect-name bind-type bind-subtype* **ERROR USING** *auth-id* **AUTHORITY PACKAGE** *package-name* **IS NOT A TRIGGER PACKAGE**

Explanation: The subcommand REBIND TRIGGER

PACKAGE can be used only for trigger packages.

bind-type
Type of bind subcommand: REBIND

bind-subtype
Subtype of bind subcommand: TRIGGER

auth-id Authorization ID attempting the REBIND

package-name
Name of the package in the following format: 'collection.package'

System action: The package specified is not rebound.

System programmer response: For a package that is not a trigger package, rebind using the subcommand REBIND PACKAGE. Re-invoke the bind subcommand with a 'collection.package' that is a trigger package.

DSNT266I *csect-name bind-type* **FOR** *object-type* = *object-name* **FAILED. THE TOTAL LENGTH OF THE PATH SCHEMA NAME LIST EXCEEDS 254 CHARACTERS WITH SCHEMA NAME** *schema-name*

Explanation: DB2 uses the schema name list you specified in the bind option PATH to set the DB2 SQL Path that will be used for the bind process. The SQL Path is a VARCHAR(254) string containing the list of schema names you specified, where each schema name is delimited with double quotes and separated from the next schema name by a comma. The total length of this SQL Path string, including the delimiters and commas, cannot exceed 254 characters. The indicated schema name *schema-name* in the bind option PATH causes the SQL Path string to exceed 254 characters.

bind-type
Type of bind subcommand: BIND or REBIND

object-type
PLAN or PACKAGE

object-name
If *object-type* is PLAN, *object-name* is the name of the plan.

If *object-type* is PACKAGE, *object-name* is the name of the package in the format 'location-id.collection-id.package-id.(version-id)'.

schema-name
The schema name specified in the bind option PATH causing the SQL Path to exceed 254 characters.

System action: The *bind-type* operation is not performed.

User response: In the bind option PATH, specify a schema name list that does not cause the SQL Path to exceed 254 characters.

For more information about the bind option PATH and the schema name list, please see: Commmand Reference.

System programmer response: See the 'User Response' section of this message.

DSNT267I *csect-name* **A TRIGGER PACKAGE CANNOT BE FREED USING THE FREE PACKAGE COMMAND**

Explanation: A FREE PACKAGE command was received that identified the trigger package to be freed. Trigger packages can only be deleted using the DROP TRIGGER SQL statement.

System action: The TSO Attachment Facility is ready for more input.

User response: Use the SQL statement DROP TRIGGER to delete the trigger package.

DSNT268I *csect-name* **A BIND COMMAND USING THE COPY OPTION NAMED A TRIGGER PACKAGE AS THE SOURCE FOR THE BIND**

Explanation: A trigger package cannot be used as the source for a BIND COPY command.

System action: The TSO Attachment Facility is ready for more input.

User response: Correct the BIND command and resubmit.

DSNT270I **THE FOLLOWING SQLCA INFORMATION WAS RETURNED FROM** *subsystem*
SQLCODE = sqlcode
SQLSTATE = sqlstate
SQLERRMT = tokens
SQLWARN warn_fields

Explanation: A warning or error has been received from a database server using Advanced SQL protocols (ASP) of the Distributed Relational Database Architecture (DRDA), during remote program BIND processing. The tokens of the message are:
1. subsystem - the database server,
2. sqlcode - SQL return code,
3. sqlstate - additional set of return codes common to the four relational products,
4. tokens - specific tokens passed in the message portion of the SQLCA for problem determination, and
5. warn_fields - the warning information indicators.

This message is issued by the following CSECT: DSNLXMSG

System action: See the System Action associated with the indicated SQL return code.

System programmer response: See the Programmer Response associated with the indicated SQLCODE and/or SQLSTATE. Take note of the warning or error (as indicated in the SQLCODE, SQLSTATE, or one or more of the SQLWARN fields), and determine if a program change is necessary. SQLWARNx fields are described in Appendix D of *DB2 SQL Reference* . A change to the application may be necessary or the system programmer may have to be consulted depending on the situation.

If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 5.

DSNT271I THE FOLLOWING SQLCA INFORMATION WAS RETURNED FROM *subsystem*

SQLCODE = sqlcode
SQLSTATE = sqlstate
SQLERRMT = tokens
SQLWARN warn_fields
STATEMENT = stmt_num

Explanation: A warning or error has been received from a database server using advanced SQL protocols (ASP) of the Distributed Relational Database Architecture (DRDA), during remote program BIND processing. The tokens of the message are:

1. subsystem - database server,
2. sqlcode - SQL return code,
3. sqlstate - additional set of return codes common to the four relational products,
4. tokens - specific tokens passed in the message portion of the SQLCA for problem determination,
5. warn_fields - the warning information indicators, and
6. stmt_num - the statement number of the failing statement.

This message is issued by the following CSECT: DSNLXMSG

System action: See the System Action associated with the indicated SQLCODE.

System programmer response: Take note of the warning or error (as indicated in the SQLCODE, SQLSTATE, or one or more of the SQLWARN fields), and determine if a program change is necessary. SQLWARNx fields are described in Appendix D of *DB2 SQL Reference* . A change to the application may be necessary or the system programmer may have to be consulted depending on the situation.

If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 5.

DSNT300I AUTH-ID *authorization-id* **NOT AUTHORIZED TO PERFORM** *function* **ON DATABASE** *dbname***. REQUEST REJECTED**

Explanation: The invoker's 'authorization-id' as designated by AUTHORIZATION ID is not authorized to perform the specified database control command function on this database. The 'function' is either -START, -STOP, or -DISPLAY. The name of the database is 'dbname'.

System action: The request is rejected.

User response: Notify the system programmer.

System programmer response: The privilege to issue the command must be granted to the indicated 'authorization-id'.

DSNT301I INVALID DATABASE *dbname*

Explanation: The databases named in the -START, -STOP, or -DISPLAY DATABASE command do not exist. The name of the database is 'dbname'.

System action: The request is rejected. This message may be repeated.

System programmer response: Reissue the command with a valid name.

DSNT302I INVALID NAME *spname*

Explanation: In the message, 'spname' is the name of the table space or index space. The 'spname' specified in the SPACENAM keyword on the -START, -STOP, or -DISPLAY DATABASE command does not exist.

System action: The request is rejected. This message may be repeated.

System programmer response: Reissue the command with a valid table space name or index name.

DSNT304I *csect-name* **INSTALLATION SYSOPR IS NOT AUTHORIZED TO CHANGE OBJECT STATUS ON DATABASE** *dbname***. REQUEST IS REJECTED.**

Explanation: INSTALLATION SYSOPR is not authorized to perform the START DATABASE command function on this database because the command can change an objects status in this database. INSTALL SYSOPR can use the START DATABASE command only to recover LPL pages or GRECP objects. The name of the database is *dbname*.

System action: The request is rejected.

Operator response: Reissue the -START DATABASE

command using the correct access mode or contact your system programmer.

DSNT305I SUBSYSTEM FAILURE DURING *command-name* DATABASE COMMAND PROCESSING. RC = *yy*, REASON = *zz*

Explanation: An error was found while processing the -START, -STOP or -DISPLAY DATABASE command. In the message, 'command-name' is the string 'DISPLAY', 'STOP', or 'START' (depending upon the command processed).

System action: Originating from CSECT DSNTDCST:
- For relational data system (RDS) subcomponent errors:
  - 'yy' is a 4-byte return code of 8.
  - 'zz' is a 4-byte abend code of '00E30085'.
- For data manager (DM) subcomponent errors:
  - 'yy' is a 4-byte return code consisting of a 2-byte return code and a 2-byte reason code returned by the data manager function invoked by DSNTDCST.
  - 'zz' is a 4-byte abend code '00E30086'.

  A record is written to SYS1.LOGREC, and an SVC dump is requested before the execution unit is abended.

Originating from CSECT DSNTDDIS:
- 'yy' is a 4-byte return code of 12.
- 'zz' is a 4-byte abend code of '00E30089'.

  The request is abended.

Originating from CSECTs DSNTDMUL or DSNTDMSG:
- 'yy' is a 4-byte return code returned by the message generator.
- 'zz' is a 4-byte abend code of '00E30084'.

  The request is abended.

System programmer response: Refer to *DB2 Codes* for the particular abend reason code issued.

Problem determination: Collect the following diagnostic items listed in Appendix A, “Problem determination,” on page 659: 1, 2, 5, 17.

DSNT306I RESPONSE MESSAGE LIMIT HAS BEEN REACHED

Explanation: A fixed amount of space exists in which to generate DISPLAY, START, or STOP database command messages. This message indicates that the DISPLAY, START, or STOP database command that was specified generated more messages than will fit in that space.

User response: Specify individual databases and/or table spaces rather than specifying '*' (that is, all names).

DSNT307I INVALID RANGE SPECIFICATION *string1* IS GREATER THAN *string2*

Explanation: A range was specified on a DISPLAY DB(string1:string2) or DISPLAY DB(..) SPACENAM(string1:string2) command or on a database command (START DB, STOP DB, or DISPLAY DB) on PART(string1:string2) keywordwhere the first string, specified alphanumerically, collates after the second string.

User response: Specify the command again. Make sure the first string does not alphanumerically collate after the second string specified.

DSNT308I AFTER KEYWORD ONLY VALID FOR A SINGLE DATABASE OR A SINGLE TABLESPACE/INDEXSPACE WITHIN A DATABASE. REQUEST REJECTED

Explanation: The AFTER keyword may be specified by one of the following:
- A single database name after the keyword DATABASE
- A single table space name or index space name after the keyword SPACENAM.

User response: Specify the command correctly.

DSNT309I *csect-name object_name* IS CURRENTLY *state*.

Explanation: One of the following is already in the requested state:
- A database listed in START DB(list) or STOP DB(list)
- A table space listed in START DB(name) SPACENAM(list) or STOP DB(name) SPACENAM(list)
- An index space listed in START DB(name) SPACENAM(list) or STOP DB(name) SPACENAM(list)
- A partition of a partitioned table space listed in START DB(list) SPACENAM(list) PART(n), STOP DB(list) SPACENAM(list) PART(n), START DB(list) SPACENAM(list), or STOP DB(list) SPACENAM(list)

In the message:
- The 'object' is the database, table space, index space or partition number with table space name or the index space name.
  - TABLESPACE, if the specified table space was currently in the requested state
  - INDEXSPACE, if the specified index space was currently in the requested state
  - DATABASE, if the database was currently in the requested state
  - PARTITION, if the specified partition of a partitioned table space or partitioned index was currently in the requested state

- The 'state' is:
  - START RW, if the specified object is currently started read/write
  - START RO, if the specified object is currently started read only
  - START UT, if the specified object is currently started for utility processing
  - STOPPED, if the specified object is currently stopped or is stopping
  - STOP PENDING, if the specified object currently is in stop pending status.

If there are pages in the logical page list (LPL) for the object, or if the object is in group buffer pool recovery pending (GRECP) status, a START DATABASE ACCESS(RO) or START DATABASE ACCESS(RW) command triggers a recovery of that object even if this message is issued for that object. For each object that has LPL or GRECP recovery that is initiated, DB2 issues a DSNI006 message followed by either a DSNI021 message (when the recovery completes successfully) or a DSNI005 message (when the recovery is unsuccessful).

User response: No response is necessary. This message is informational only.

DSNT310I *csect-name* **ALL** *object* **CURRENTLY** *state***.**

Explanation: A database and all of its table spaces and index spaces are already in the requested state. This occurred following the processing of one of these commands:
- START DB(*)
- STOP DB(name)
- START DB(name) SPACENAM(*)
- STOP DB(name) SPACENAM(*).

In this message:
- The 'object' is either:
  - TABLESPACE(S) if the specified table space was currently in the requested state
  - INDEXSPACE(S) if the specified index space was currently in the requested state
  - DATABASE(S) if the database was currently in the requested state.
- The 'state' is either:
  - START RW if the specified table space is currently started read/write
  - START RO if the specified table space is currently started read only
  - START UT if the specified table space is currently started for utility processing
  - STOPPED if the specified table space is currently stopped, has a stop pending against it, or is stopping.

If there are pages in the logical page list (LPL) for the object, or if the object is in group buffer pool recovery pending (GRECP) status, a START DATABASE ACCESS(RO) or START DATABASE ACCESS(RW) command triggers a recovery of that object even if this message is issued for that object. For each object that has LPL or GRECP recovery that is initiated, DB2 issues a DSNI006 message followed by either a DSNI021 message (when the recovery completes successfully) or a DSNI005 message (when the recovery is unsuccessful).

User response: No response is necessary. This message is informational only.

DSNT311I MESSAGE LIMIT EXCEEDED. DISPLAY IS TERMINATED.

Explanation: The number of messages generated exceeds the maximum number of messages specified in the LIMIT keyword or exceeds the system default.

DSNT312I *object name* **MUST BE STARTED RW. NO ACTION TAKEN.**

Explanation: The ACCESS keyword is either RO or UT, and the object must be started ACCESS (RW).

In the message, 'object' is either:
- TABLESPACE(S) IN if the START DB(name) SPACENAM(...) is specified, or
- DATABASE if the START DB(name) is specified.

DSNT313I ACCESS(FORCE) IS VALID ONLY WITH EXPLICIT TABLESPACE OR INDEXSPACE NAMES. COMMAND IS REJECTED

Explanation: The ACCESS(FORCE) keyword can be specified when either:
- A single table space or index space name appears after the keyword SPACENAM in a START DATABASE command.
- A list of table space and/or index space names appears after the keyword SPACENAM in a START DATABASE command.

Operator response: Specify the command correctly.

DSNT314I *csect_name* **INVALID PARTITION SPECIFICATION FOR OBJECT** *object-name*

Explanation: The PART keyword is invalidly specified in a START DATABASE, STOP DATABASE, or DISPLAY DATABASE command.

The PART keyword is invalid if it is specified for:
- A nonpartitioned table space
- A nonpartitioned type 1 index
- A type 2 index not defined on a partitioned table space

The object *object-name* can be either a table space or an index space.

No action for this table space or index space is performed by the START DATABASE, STOP DATABASE or DISPLAY DATABASE command.Any other valid table space or index space specified is processed by the command.

Severity: 8 (error)

System action: The desired command is not processed for the table space or index space in question. All other valid table spaces or index spaces specified in the command are processed.

User response: Resubmit the command for the nonpartitioned table space or index space without using the PART keyword.

Problem determination: SYSIBM.SYSTABLESPACE in the DB2 catalog indicates whether a table space is partitioned. SYSIBM.SYSINDEXSPACE in the DB2 catalog indicates whether an index space is partitioned and whether the index space is a type 1 or a type 2 index. Type 2 nonpartitioned indexes can be started, stopped, or displayedat the logical partition level; type 1 cannot.

DSNT315I *csect_name* **INVALID PARTITION NUMBER OR RANGE SPECIFIED -** *part-num***FOR OBJECT** *object-name*

Explanation: An invalid partition number or rangewas specified in the PART keyword for a START DATABASE, STOP DATABASE or DISPLAY DATABASE command.

Partition number or range *part-num*is invalid for the object *object-name*.

Severity: 8 (error)

System action: The part number or rangeis ignored. All other valid partitions specified are processed if PART was specified as list.

User response: Determine the correct partition specification for the table space or index that was the subject of the specified command and resubmit the command.

Problem determination: SYSIBM.SYSTABLESPACE in the DB2 catalog indicates the correct range of partition numbers for the table space. SYSIBM.SYSINDEXSPACE in the DB2 catalog indicates whether an index space is partitioned and whether the index space is a type 1 or a type 2 index. Type 2 indexes can be started, stopped, or displayedat the logical partition level; type 1 cannot.

DSNT316I *csect-name* **A LOGICAL PARTITION OF AN INDEX WHICH HAS PAGES IN THE LPL OR HAS BEEN MARKED GRECP CANNOT BE STARTED FOR RW OR RO ACCESS**

Explanation: The START DATABASE command was specified for a logical partition of a nonpartitioned index which has pages in the logical page list (LPL) or is marked group buffer pool recovery pending (GRECP). To start a nonpartitioned index with one of these exception conditions, the START DB command must be specified without the PART keyword.

Severity: 8 (error)

System action: The command is not processed for the index space. All other valid table spaces or index spaces specified in the command are processed.

User response: Submit the command for the nonpartitioned index space without using the PART keyword.

Problem determination: If more information on the status of the index space is required, use the -DISPLAY DATABASE command.

DSNT317I *csect-name* **ANY USER-DEFINED INDEXES ON DSNDB06 COULD NOT BE** *cmd-type* **DUE TO A RESOURCE UNAVAILABLE ON THE CATALOG OR DIRECTORY.**

Explanation: If user-defined indexes exist on DSNDB06, they could not be displayed, started or stopped (depending on *cmd-type*) because the catalog or directory was not available. DB2-defined indexes and table spaces are not affected by this unavailability.

Programmer response: When you get this message, correct the resource unavailable problem and reissue the command.

System action: Any user-defined indexes on DSNDB06 were not affected by the command issued. All other table spaces and index spaces were processed by the command.

DSNT318I *csect-name* **PLAN=***plan-id1* **WITH CORRELATION-ID=***id1* **CONNECTION-ID=***id2* **LUW-ID=***id3* **THREAD-INFO=***id4* **TIMED OUT BECAUSE OF A P-LOCK HELD BY MEMBER** *id5*

Explanation: A plan cannot get an IRLM lock because the resource is held by a P-lock in the data sharing group, and the amount of time to wait for the locked resource was exceeded.

The following identifiers are used in the message:

plan-id1
The name of the executing plan.

id1 The correlation identifier (ID) of the thread associated with plan *plan-id1*. Correlation IDs beginning with '0nn', where 'nn' is a number from 01 to 28, identify system agents.

id2 The connection-ID of plan *plan-id1*.

id3 The logical-unit-of-work identifier ('LUW-ID') of *plan-id1*. An '*' indicates that the thread does not have a LUW-ID. If the LUW-ID is not an '*', the value is a logical-unit-of-work-id that is assigned to the thread followed by an "=" sign and a token value, which can be used in place of logical-unit-of-work-id in any DB2 command that accepts logical-unit-of-work-id as input.

id4 This ID contains additional identifiers that define more specific information regarding the identity of the agent.

In many distributed configurations, the primary authorization-ID used with DB2 is not necessarily the user-ID of the end-user. If provided by the client, the additional tokens further assist in identifying the client end-user.

The following identifiers, separated by a colon (:), are provided:

- The primary authorization-ID associated with the thread.
- Workstation name of the end-user's workstation. An asterisk (*) indicates it is not relevant to the thread or was not provided by the client.
- End-user's user-ID. An asterisk (*) indicates it is not relevant to the thread or was not provided by the client.
- Application name of the end-user's application. An asterisk (*) indicates that is not relevant to the thread or was not provied by the client.

id5 Identifies the member name of the DB2 subsystem which holds the conflicting P-lock.

System action: The SQLCA corresponding to correlation-ID *id1*, connection-ID *id2*, and LUW-ID *id3* contains an SQLCODE -904, -911 , -913 or -923 and the name of the resource involved in the timeout.

System programmer response: Identify the member holding the P-lock and determine why the P-lock is not being released. If the problem persists and no other method of releasing the P-lock can be found, consider recycling the member of the group holding the P-lock.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 4.

DSNT331I *csect-name***DATABASE** *name* **AND ANY SPACE IT CONTAINS CANNOT BE STOPPED. IT IS THE ACTIVE COMMUNICATIONS DATABASE**

Explanation: The database specified in the message cannot be stopped because it is the active communications database. In addition, spaces within the active communications database cannot be stopped.

System action: Nothing is stopped.

Operator response: If the database or any space within it must be stopped, you must stop the distributed data facility (STOP DDF). In a DB2 data sharing environment, you must stop the distributed data facility on all members of the DB2 data sharing group.

DSNT332I TABLESPACES AND INDEXSPACES ARE UNAVAILABLE BECAUSE DATABASE name IS STOPPED

Explanation: The database 'name' is currently marked as stopped. Consequently, its table spaces and index spaces are unavailable even though they may have been started.

DSNT350I –luw-id ACCESSING DATA FOR location

Explanation: This message follows message DSNT394I or DSNT396I for each thread reported that is participating in distributed data access (database access thread). This additional line reports the distributed identification as follows:

luw-id=token
The logical-unit-of-work identifier (LUW-ID) is a unique identifier that is assigned to the thread followed by an '=' sign and a token. The token can be used in place of luw-id in any DB2 command that accepts luw-id as input.

location
The LOCATION name assigned to the requesting site.

The message number and subsystem recognition character are not included in this message.

This message is issued by the following CSECT: DSNTDMUL

DSNT358I *csect_name* **INVALID PARTITION SPECIFICATION FOR OBJECT** *object-name*

Explanation: Specification of the PART keyword in a DISPLAY DATABASE command is invalid.

The PART keyword is invalid if it is specified for:

- A nonpartitioned table space
- A nonpartitioned type 1 index
- A type 2 index not defined on a partitioned table space

The object *object-name* can be either a table space or an index space.

No action for this table space or index space is performed by the DISPLAY DATABASE command. Any other valid table space or index space specified is processed by the command.

Severity: 8 (error)

System action: The desired command is not processed for the table space or index space in question. All other valid table spaces or index spaces specified in the command are processed.

User response: Resubmit the command for the nonpartitioned table space or index space without using the PART keyword.

Problem determination: SYSIBM.SYSTABLESPACE indicates whether a table space is partitioned. SYSIBM.SYSINDEXSPACE indicates whether an index space is partitioned and whether the index space is a type 1 or a type 2 index. Type 2 nonpartitioned indexes can be displayed at the logical partition level; type 1 indexes cannot.

DSNT359I *csect_name* **INVALID PARTITION NUMBER SPECIFIED -** *part-num* **FOR OBJECT** *object-name*

Explanation: An invalid partition number was specified in the PART keyword for a DISPLAY DATABASE command.

Partition number *part-num* is invalid for the object *object-name*.

Severity: 8 (error)

System action: The part number is ignored. All other valid partitions specified are processed.

User response: Determine the correct partition specification for the table space or index that was the subject of the specified command and resubmit the command.

Problem determination: SYSIBM.SYSTABLESPACE in the DB2 catalog indicates the correct range of partition numbers for the table space. SYSIBM.SYSINDEXSPACE in the DB2 catalog indicates whether an index space is partitioned and whether the index space is a type 1 or a type 2 index. The PART keyword can be used to display the status of logical partitions of type 2 indexes but not type 1 indexes.

DSNT360I **********

Explanation: This message consists of 35 asterisks that begin the display of any database and the display of space names for any database.

DSNT361I ***DISPLAY DATABASE SUMMARY* report-type-list**

Explanation: This message is a multiple-line message that labels the requested set of databases, table spaces, and index spaces.

In the message, 'report-type-list' lists all options specified on the command delimited by spaces. Possible values are:

ACTIVE
All allocated databases are listed.

GLOBAL
All requested database information is listed. This title is produced if ACTIVE, DISABLED, and RESTRICT are omitted.

LOCKS
The connection identifiers and correlation identifiers of all applications holding, retaining, or waiting for locks on the specified table spaces or index spaces are displayed.

RESTRICT
All databases with restricted usage (that is, RO, started UT) are listed.

USE
The connection identifiers and correlation identifiers of all allocated applications are listed.

CLAIMERS
The connection identifiers and correlation identifiers of all applications holding claims on the specified database are displayed.

LPL
The page number ranges of the pages that are included in the logical page list for the table space or index space.

OVERVIEW
All table spaces and index spaces that are contained within the databased that are specified.

DSNT362I **DATABASE =** *data-base-name* **STATUS =** *status* **DBD LENGTH =** *dbd-length*

Explanation: This message signals the beginning of display information from the DISPLAY DATABASE command. It lists the name, status, and DBD length of the database. All table space and index space names listed in the following DSNT392I, DSNT394I, or DSNT396I messages are contained within this database.

data-base-name
The database name.

status A combination of one or more of the following strings delimited by a comma. The maximum length of the string is 18 characters.

RO The database is started for read activity only.

RW The database is started for read and write activity.

STOP The database is stopped.

STOPP A stop is pending or is in progress for this database.

UT The database is started for utility processing only.

DSNT363I PHYSICAL ERROR PAGE RANGE FOR DATA SET data-set-name LOW=lowphy, HIGH=hiphy

Explanation: This message is the response to a -DISPLAY DATABASE command when the SPACENAM keyword is specified in order to notify the user of any physical I/O errors encountered on a data set.

In the message, 'data-set-name' is the name of the data set with errors. The variables 'lowphy' and 'hiphy' identify the range of physical pages, within the data set, that were being read when the I/O errors occurred. Physical page numbers are assigned in relationship to the beginning of the data set in which the pages reside.

This message is always preceded by messages DSNT360I and DSNT361I, which identify the name of the database and page set that the specified data set supports.

Page sets (table spaces or index spaces) are made up of supporting data sets. Physical pages are pages in supporting data sets that are numbered from the beginning of the particular data set in which they reside. This message refers to physical pages in a data set that support (are part of) some page set.

System action: See the Explanation section for this message.

DSNT364I *csect-name* **KEYWORD** *ONLY* **IS USED INCORRECTLY**

Explanation: Keyword *ONLY* is used incorrectly in the DISPLAY DATABASE command. When the SPACENAM keyword is specified, keyword *ONLY* is valid only if one of the following keywords is also used:

LOCKS
LPL
CLAIMERS
WEPR
USE

System action: The DISPLAY DATABASE command is not processed.

User response: Add one of the keywords mentioned above in the explanation to the DISPLAY DATABASE command and issue the command again.

DSNT365I NO DATABASES FOUND

Explanation: This message is produced when a DISPLAY DB(name) is entered and no databases were found to satisfy the request.

System action: The process continues normally.

DSNT366I DATABASE name NOT type

Explanation: A database listed in the DISPLAY DB(...) is not of the correct type. The variables are as follows:

name The name of the object.

type The type of request specified: ACTIVE, RESTRICTED, ACTIVE OR RESTRICTED

Note that either DSNT301I or DSNT302I is produced if no object is found and the keywords ACTIVE and RESTRICT are omitted.

System action: Processing continues normally.

DSNT368I *csect-name* **MORE THAN ONE MUTUALLY EXCLUSIVE KEYWORD WAS SPECIFIED IN THE DISPLAY DATABASE COMMAND.**

Explanation: The keywords, LOCKS, LPL, CLAIMERS, WEPR and USE are mutually exclusive. Only one should be specified in a DISPLAY DATABASE command. The precedence order for which keyword takes precedence over the others is:

1. LPL
2. CLAIMERS
3. LOCKS
4. USE
5. WEPR

System action: The DISPLAY DATABASE command is processed with this warning message.

User response: Remove keywords from DISPLAY DATABASE command until only one of these mutually exclusive keywords is specified.

DSNT367I NO INFORMATION AVAILABLE

Explanation: This message indicated there is no LOCK, USE, LPL, WEPR or CLAIMER information available when DISPLAY DB(...) SPACENAM(...) ONLY is entered with the keywords LOCKS, USE, LPL, WEPR or CLAIMERS.

System action: The process continues normally.

DSNT375I **PLAN=***plan-id1* **WITH CORRELATION-ID=***id1* **CONNECTION-ID=***id2* **LUW-ID=***id3* **THREAD-INFO=***id4* **IS DEADLOCKED WITH PLAN=***plan-id2* **WITH CORRELATION-ID=***id5* **CONNECTION-ID=***id6* **LUW-ID=***id7* **THREAD-INFO=** *id8* **ON MEMBER** *id9*

Explanation: This message is issued whenever plan *plan-id1* with correlation *id1*, connection *id2*, and LUW *id3* is denied an IRLM lock because of a detected deadlock. Plan *plan-id2* identifies one of the members of the deadlock.

DB2 does not attempt to identify all survivors of a deadlock or all participants in a deadlock in the DSNT375I message. Plan *plan-id2* in message DSNT375I might be just one of several plans holding locks on the desired resource.

If multiple deadlocks are detected during the same deadlock-detection interval, the lock holder identified by *plan-id2* in one DSNT375I message might be selected to break a deadlock identified by a different DSNT375I message. The lock holder identified by *plan-id2* is not guaranteed to be a winner in all deadlocks in which it participates.

A DB2 IFCID 172 trace record contains the information needed to determine all participants in, and survivors of, a given deadlock.

plan-id1, plan-id2
The names of the executing plans.

id1 The correlation identifier of the thread associated with *plan-id1*. Correlation IDs beginning with '0nn', where 'nn' is a number from 01 to 28, identify a system agent.

id2 The connection identifier of the thread associated with *plan-id1*.

id3 The logical-unit-of-work identifier of the thread associated with *plan-id1*. An asterisk (*) indicates that the thread does not have an LUW-ID. If the LUW-ID is not an asterisk, the value is a logical-unit-of-work identifier that is assigned to the thread. It is followed by an equal (=) sign and a token value, which can be used in place of the logical-unit-of-work identifier in any DB2 command that accepts a logical-unit-of-work identifier as input.

id4 This ID contains additional identifiers that define more specific information regarding the identity of the agent.

In many distributed configurations, the primary authorization-ID used with DB2 is not necessarily the user-ID of the end-user. If provided by the client, the additional tokens further assist in identifying the client end-user.

The following identifiers, separated by a colon (:), are provided:

- The primary authorization-ID associated with the thread.
- Workstation name of the end-user's workstation. An asterisk (*) indicates it is not relevant to the thread or was not provided by the client.
- End-user's user-ID. An asterisk (*) indicates it is not relevant to the thread or was not provided by the client.
- Application name of the end-user's application. An asterisk (*) indicates that is not relevant to the thread or was not provided by the client.

id5 The correlation identifier of the thread associated with *plan-id2*.

id6 The connection identifier of the thread associated with *plan-id2*.

id7 The logical-unit-of-work identifier of the thread associated with *plan-id2*. An asterisk (*) indicates that the thread does not have an LUW-ID. If the LUW-ID is not an asterisk, the value is a logical-unit-of-work-identifier that is assigned to the thread followed by an equal (=) sign and a token value, which can be used in place of logical-unit-of-work identifier in any DB2 command that accepts logical-unit-of-work identifier as input.

id8 This ID contains additional identifiers that define more specific information regarding the identity of the agent.

In many distributed configurations, the primary authorization-ID used with DB2 is not necessarily the user-ID of the end-user. If provided by the client, the additional tokens further assist in identifying the client end-user.

The following identifiers, separated by a colon (:), are provided:

- The primary authorization-ID associated with the thread.
- Workstation name of the end-user's workstation. An asterisk (*) indicates it is not relevant to the thread or was not provided by the client.
- End-user's user-ID. An asterisk (*) indicates it is not relevant to the thread or was not provided by the client.
- Application name of the end-user's application. An asterisk (*) indicates that is not relevant to the thread or was not provided by the client.

id9 Identifies the member name of the DB2 subsystem on which the thread associated with *plan-id2* is executing.

DB2 can recover from a lock denial in some cases and can complete successfully. In such a case, the SQLCA is not filled in. One example is during setup of performance enhancements during bind time for execution time.

The SQLCA corresponding to correlation id *id1*, connection id *id2*, and LUW-ID *id3* contains an SQLCODE of -904, -911, -913, or -923 and the name of the resource involved in the deadlock.

If you receive a system plan in this message, see the information about system plans in message DSNT376I.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 5, 18.

```
# DSNT376I   PLAN=plan-id1 WITH
#            CORRELATION-ID=id1
#            CONNECTION-ID=id2 LUW-ID=id3
#            THREAD-INFO=id4 IS TIMED OUT.
#            ONE HOLDER OF THE RESOURCE IS
#            PLAN=plan-id2 WITH
#            CORRELATION-ID=id5
#            CONNECTION-ID=id6 LUW-ID=id7
#            THREAD-INFO= id8 ON MEMBER id9
```

Explanation: This message is issued whenever plan *plan-id1* with correlation *id1*, connection *id2*, and LUW *id3* is denied a lock by IRLM, because the amount of time to wait for a locked resource was exceeded. Plan *plan-id2* identifies one of the plans holding a lock on the desired resource. The listed plan is not necessarily responsible for the time-out condition.

plan-id1, plan-id2
The names of the executing plans.

id1 The correlation identifier of the thread associated with *plan-id1*. If the execution of the plan was switched by DB2 to a system agent, the correlation identifier identifies the system agent. Correlation IDs beginning with '0nn', where 'nn' is a number from 01 to 28, identify system agents.

id2 The connection identifier of the thread associated with *plan-id1*. If the execution of the plan was switched by DB2 to a system agent, the connection identifier is the DB2 subsystem name.

id3 The logical-unit-of-work identifier of the thread associated with *plan-id1*. An '*' indicates that the thread does not have a LUW-ID. If the LUW-ID is not an '*', the value is a logical-unit-of-work-ID that is assigned to the thread followed by an "=" sign and a token value, which can be used in place of logical-unit-of-work-ID in any DB2 command that accepts logical-unit-of-work-ID as input.

id4 This ID contains additional identifiers that define more specific information regarding the identity of the agent.

In many distributed configurations, the primary authorization-ID used with DB2 is not necessarily the user-ID of the end-user. If provided by the client, the additional tokens further assist in identifying the client end-user.

The following identifiers, separated by a colon (:), are provided:

- The primary authorization-ID associated with the thread.
- Workstation name of the end-user's workstation. An asterisk (*) indicates it is not relevant to the thread or was not provided by the client.
- End-user's user-ID. An asterisk (*) indicates it is not relevant to the thread or was not provided by the client.
- Application name of the end-user's application. An asterisk (*) indicates that is not relevant to the thread or was not provided by the client.

id5 The correlation identifier of the thread associated with *plan-id2*. If the execution of the plan was switched by DB2 to a system agent, the correlation identifier identifies the system agent. Correlation IDs beginning with '0nn', where 'nn' is a number from 01 to 28, identify system agents.

id6 The connection identifier of the thread associated with *plan-id2*. If the execution of the plan was switched by DB2 to a system agent, the connection identifier is the DB2 subsystem name.

id7 The logical-unit-of-work identifier of the thread associated with *plan-id2*. An '*' indicates that the thread does not have a LUW-ID. If the LUW-ID is not an '*', the value is a logical-unit-of-work-ID that is assigned to the thread followed by an "=" sign and a token value, which can be used in place of logical-unit-of-work-ID in any DB2 command that accepts logical-unit-of-work-ID as input.

id8 This ID contains additional identifiers that define more specific information regarding the identity of the agent.

In many distributed configurations, the primary authorization-ID used with DB2 is not necessarily the user-ID of the end-user. If provided by the client, the additional tokens further assist in identifying the client end-user.

The following identifiers, separated by a colon (:), are provided:

- The primary authorization-ID associated with the thread.
- Workstation name of the end-user's workstation. An asterisk (*) indicates it is not relevant to the thread or was not provided by the client.
- End-user's user-ID. An asterisk (*) indicates it is not relevant to the thread or was not provided by the client.
- Application name of the end-user's application. An asterisk (*) indicates that is not relevant to the thread or was not provided by the client.

id9 Identifies the member name of the DB2 subsystem on which the thread associated with *plan-id2* is executing.

DB2 can recover from a lock denial in some cases and can complete successfully. In such a case, the SQLCA is not filled in. One example is during setup of performance enhancements during bind time for execution time.

The SQLCA corresponding to correlation-ID *id1*, connection-ID *id2*, and LUW-ID *id3* contains an SQLCODE of -904 or -923 and the name of the resource involved in the timeout.

Several different system plans can appear in *plan-id1* and *plan-id2*. These system plans are not documented in SYSIBM.SYSPLAN. If you receive a system plan in this message, you are in contention with another user. If you are not doing an automatic rebind, message DSNT500I or DSNT501I might accompany this message and identify that resource. If you need additional information, refer to the SQLCA and any additional bind messages. If you are doing an auto rebind, the SQLCA is not available. Refer to Part 5 of *DB2 Application Programming and SQL Guide* for more information about automatic rebinding. The system plans are:

ACT
BCT
DISTSERV
DSNBIND
STARTCT
DSNUTIL.

Operator response: Notify the system programmer.

System programmer response: Timeouts can happen for many reasons, including factors relating to DB2 or IRLM. Some of those reasons are listed below along with possible ways to reduce or eliminate your timeout. Refer to Part 5 (Volume 2) of *DB2 Administration Guide*for more detail about the concepts discussed in this list. See also SQLCODEs -911 and -913.

Timeouts can result from the definition of objects or the structure of applications as explained below.

- If the resource involved in the timeout is listed in the DSNT500I/DSNT501I message as a table or table space, check the following items for other possible causes of timeout:
  1. Check the LOCKRULE column value in SYSIBM.SYSTABLESPACES for the table space being accessed. A lock size of TABLE or TABLESPACE could cause a timeout. If your application does not need to lock the entire table or table space, you can resolve the timeout by changing the lock size to PAGE or ANY using the ALTER statement.
  2. Check the number of LOCKS PER TABLE(SPACE), which was set at the time DB2 was installed. If a large number of page or row locks are acquired and held, a small value for LOCKS PER TABLE(SPACE) could be causing lock escalation to a table or table space lock. The timeout can be resolved by changing the value on the DSNTIPK update panel or by binding the application's plan with RELEASE(COMMIT) and doing commits more frequently.
  3. You can use EXPLAIN (described in Part 5 of *DB2 Application Programming and SQL Guide*) with the failing SQL statement and check the TSLOCKMODE column for every table/table space in the EXPLAIN output. If the competing applications are attempting to obtain incompatible locks, you might be able to resolve the timeout by running the applications sequentially rather than concurrently or by following the suggestions in other steps listed here, such as changing lock size, ACQUIRE/RELEASE parameters or ISOLATION type.
  4. Check the ACQUIRE/RELEASE values specified on the plan. ACQUIRE(ALLOCATE) acquires all locks for all tables and table spaces referenced in the plan, even if the program logic does not cause the table/table space to be accessed. Binding with RELEASE(DEALLOCATE) causes table, table space and DBD locks to be held longer than binding with RELEASE(COMMIT), possibly causing a timeout.
  5. Use the DISPLAY DB LOCKS command during program execution to verify that the locks being held at the time of the timeout are as expected. If they are not, you may need to use ALTER to change the locksize of the table space if possible, or rebind your plan with different ACQUIRE, RELEASE or ISOLATION values.
  6. ISOLATION level affects the mode of table/table space locks chosen and may be of concern if the plan is bound with RR and there is a page set scan.
- Check the ISOLATION level specified on the plan in SYSIBM.SYSPLAN. A value of RR is more likely to cause a timeout because it requires more page or row

locks and also has a greater potential than CS for causing lock escalation, which can cause timeouts as noted above.
- Check the frequency of COMMITs in the application. A large number of updates without any COMMITs can cause a timeout because the more pages that are changed, the more page or row locks held, and the greater potential for lock escalation.

This condition might be caused by an undetected deadlock, for example, an agent holding a lock on the resource for a period of time that exceeds either the specified time or the IRLM default wait time.

If the time limit is too small, you can increase the limit by specifying a new IRLM locked resource wait time limit. To do this, you can either use the parameter in the DSN6SPRM assembler macro in the DSNTIJUZ job stream or update the 'wait-time' field of the DB2 installation IMS Resource Lock Manager panel, IRLMRWT. You must also specify all other parameters contained in this version of the CSECT. Then do one of the following:
- Resubmit installation job DSNTIJUZ with the link-edit SYSIN file 'NAME' parameter that matches the -START DB2 'PARM=' parameter.
- Reassemble DSN6SPRM and relink-edit DSNZPARM by resubmitting installation job DSNTIJUZ. The link-edit SYSIN file 'NAME' parameter must match the -START DB2 'PARM=' parameter.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 5, 9, 18.

DSNT377I **PLAN=***plan-id1* **WITH CORRELATION-ID=***id1* **CONNECTION-ID=***id2* **LUW-ID=***id3* **THREAD-INFO=***id4* **IS IN CONFLICT WITH AN INDOUBT THREAD ON MEMBER** *id5*

Explanation: This message is issued by the service controller subcomponent. A plan cannot gain an IRLM lock because a required resource is currently undergoing recovery. The message identifies the plan in this way:

plan-id1
The name of the executing plan.

id1 The correlation identifier (ID) of the thread associated with plan *plan-id1*. Correlation IDs beginning with '0nn', where 'nn' is a number from 01 to 28, identify system agents.

id2 The connection-ID of plan *plan-id1*.

id3 The logical-unit-of-work identifier ('LUW-ID') of *plan-id1*. An '*' indicates that the thread does not have a LUW-ID. If the LUW-ID is not an '*', the value is a logical-unit-of-work ID that is assigned to the thread followed by an "=" sign and a token value, which can be used in place of logical-unit-of-work-id in any DB2 command that accepts logical-unit-of-work-id as input.

id4 This ID contains additional identifiers that define more specific information regarding the identity of the agent.

In many distributed configurations, the primary authorization-ID used with DB2 is not necessarily the user-ID of the end-user. If provided by the client, the additional tokens further assist in identifying the client end-user.

The following identifiers, separated by a colon (:), are provided:
- The primary authorization-ID associated with the thread.
- Workstation name of the end-user's workstation. An asterisk (*) indicates it is not relevant to the thread or was not provided by the client.
- End-user's user-ID. An asterisk (*) indicates it is not relevant to the thread or was not provided by the client.
- Application name of the end-user's application. An asterisk (*) indicates that is not relevant to the thread or was not provied by the client.

id5 Identifies the member name of the DB2 subsystem on which the indoubt thread is executing.

The SQLCA corresponding to correlation-ID *id1*, connection-ID *id2*, and LUW-ID *id3* contains an SQLCODE -904, -911, -913, or -923 and the name of the resource currently being recovered by DB2.

If you receive a system plan in this message, see the information about system plans in message DSNT376I.

System action: Processing continues.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 4.

DSNT378I *csect-name* **PLAN=***plan-id1* **WITH CORRELATION-ID=***id1* **CONNECTION-ID=***id2* **LUW-ID=***id3* **THREAD-INFO=***id4* **TIMED OUT BECAUSE OF A RETAINED LOCK HELD BY MEMBER** *id5*

Explanation: A plan cannot get an IRLM lock because the resource is held by a retained lock on behalf of another member in the data sharing group, and the amount of time to wait for the locked resource was exceeded. The member holding the retained lock is

either failed, or is in the process of restarting to recover its retained locks. This message is only issued if the RETLWAIT installation parameter in DSN6SPRM is set to 'YES'. If RETLWAIT is set to 'NO' (this is the default), then any IRLM lock that conflicts with a retained lock will not wait, but instead will be immediately rejected (see reason codes 00C900BE and 00C20254).

The following identifiers are used in the message:

plan-id1
The name of the executing plan.

id1 The correlation identifier (ID) of the thread associated with plan *plan-id1*. Correlation IDs beginning with '0nn', where 'nn' is a number from 01 to 28, identify system agents.

id2 The connection-ID of plan *plan-id1*.

id3 The logical-unit-of-work identifier ('LUW-ID') of *plan-id1*. An '*' indicates that the thread does not have a LUW-ID. If the LUW-ID is not an '*', the value is a logical-unit-of-work id that is assigned to the thread followed by an "=" sign and a token value, which can be used in place of logical-unit-of-work-id in any DB2 command that accepts logical-unit-of-work-id as input.

id4 This ID contains additional identifiers that define more specific information regarding the identity of the agent.

In many distributed configurations, the primary authorization-ID used with DB2 is not necessarily the user-ID of the end-user. If provided by the client, the additional tokens further assist in identifying the client end-user.

The following identifiers, separated by a colon (:), are provided:

- The primary authorization-ID associated with the thread.
- Workstation name of the end-user's workstation. An asterisk (*) indicates it is not relevant to the thread or was not provided by the client.
- End-user's user-ID. An asterisk (*) indicates it is not relevant to the thread or was not provided by the client.
- Application name of the end-user's application. An asterisk (*) indicates that is not relevant to the thread or was not provied by the client.

id5 Identifies the member name of the failed or restarting DB2 subsystem which holds the conflicting retained lock.

System action: The SQLCA corresponding to correlation-ID *id1*, connection-ID *id2*, and LUW-ID *id3* contains an SQLCODE -904, -911 , -913 or -923 and the name of the resource involved in the timeout.

System programmer response: Restart the member holding the retained lock if it has not yet been restarted.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 4.

DSNT379I CANNOT IDENTIFY TO IRLM. INVALID SSNAME OR IRLM IS NOT CURRENTLY ACTIVE

Explanation: Either an attempt was made to identify the IRLM with a subsystem name not recognized by MVS, or the IRLM is not currently active. This message is also issued if common service area (CSA) storage is insufficient.

System action: DB2 abends with a reason code of X'04E'.

Operator response: Notify system programmer.

System programmer response: This is probably a user error. Change the IRLMSID parameter in the DB2 DSN6SPRM initialization macro to match the MVS subsystem table, or start IRLM if IRLM is not already up. See return code 00E30079 for further information about determining the cause of the error.

DSNT380I CANNOT IDENTIFY TO IRLM, SSNAME ALREADY IN USE

Explanation: Either DB2 previously terminated and did not successfully sign off from IRLM, or another subsystem is identified to IRLM with this installation-defined SSNAME. This message accompanies abend reason code X'00E30071'.

System action: DB2 is abended with an abend code of X'04E'.

Operator response: Notify the system programmer.

System programmer response: Either manually stop IRLM, or change the installation-defined SSNAME.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 5, 9, 18.

DSNT381I IRLM QUIT REQUEST FAILED, RC = xx , SC = yy

Explanation: During the QUIT (or signoff) request to the IMS/VS Resource Lock Manager, a return code and subcode were returned indicating an unsuccessful completion of the request. Refer to *DB2 Codes*, for IRLM return codes and subcode meanings.

System action: This message indicates that the DB2 subsystem shutdown could not be completed normally.

Operator response: Notify the system programmer.

System programmer response: Determine the cause of

the failure from the return code and subcode, and correct the error. Refer to *DB2 Codes*, for IRLM return codes and subcode meanings. DB2 has terminated but has not signed off from IRLM. It will be necessary to issue the 'F irlmproc, ABDUMP' command to stop the IRLM. This must be done before issuing another -START DB2 command.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 5, 9, 18.

DSNT382I IRLM AUTO-START FAILED

Explanation: During the IDENTIFY to IRLM, the AUTO-START of IRLM using SVC 34 failed. This message accompanies abend code X'00E30080'.

System action: DB2 is terminated with an abend code of X'04E'.

Operator response: Notify the system programmer.

System programmer response: Determine the cause of failure from the SVC 34 return code preserved in register 2.

DSNT383I *csect-name* **IRLM AUTO-STOP FAILED**

Explanation: During a sign off from IRLM caused by either DB2 normal shutdown (CSECT DSNTLQUI) or DB2 subsystem termination (CSECT DSNTLCLE), an AUTO-STOP of IRLM using SVC 34 failed. If this occurred during normal sign off, this message accompanies abend reason code X'00E30075'.

System action: DB2 abends.

Operator response: Notify the system programmer.

System programmer response: Manually stop IRLM.

Problem determination: Determine the cause of failure from the SVC return code preserved in register 2.

Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 5, 9.

DSNT384I IRLM AUTO-START WAIT TIME EXCEEDED

Explanation: An attempt was made to automatically start IRLM. However, the wait time specified was exceeded.

System action: DB2 is abended.

Operator response: Notify the system programmer.

System programmer response: Either manually start IRLM or modify the installation-defined auto-start wait time. Restart DB2.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 5, 9.

Check the IRLMID given in the IRLM procedure against the IRLMSID in macro DSN6SPRM which was in the DSNZPARM used at startup.

DSNT386I NAME TYPE PARTS

Explanation: This message forms the headings of a display for a set of messages. This message is always followed by multiple DSNT387I messages. These DSNT387I messages are used to display information that is received as a result of a DISPLAY DATABASE command when the keyword OVERVIEW is specified. Refer to message DSNT387I for a description of the data these headings identify.

DSNT387I *space type parts*

Explanation: This message contains the results from a DISPLAY DATABASE command when the keyword OVERVIEW is specified. This message contains data that describes the status of a tablespace or index space in the database that is listed in the preceding DSNT362I message. The data in this message is always preceded by the column headings in message DSNT386I. The message number for this message is not displayed. However, messages DSNT386I, DSNT387I, and DSNT390I are always produced with a multiple line WTO and are never interrupted.

space Tablespace name, index space name, or table OBID.

type TS for a tablespace, IX for an index space, LS for a LOB tablespace, or TB for a table

parts Indicates the number of partitions in the tablespace, index space, or LOB tablespace. It is blank for a simple tablespace, simple index space, or segmented tablespace.

DSNT388I NAME TYPE PART STATUS CONNID CORRID CLAIMINFO

Explanation: This message is the heading for a set of messages that display information as a result of a -DISPLAY DATABASE command when the keyword CLAIMERS is specified. The message number for this message is not displayed. This message is always followed by multiple DSNT389I messages containing the column data. Refer to message DSNT389I for a description of the data these headings identify.

DSNT389I *space type part status conn-id corr-id claims*

Explanation: This message contains the display information that results from a -DISPLAY DATABASE command when the keyword CLAIMERS is specified. This message contains the column data describing the status of a table space or index space in the database listed in message DSNT362I.

space The table space name or index space name.

type TS for a table space, IX for an index space, LS for a LOB table space.

part The partition number. For a nonpartitioned table space or an index on a nonpartitioned table space, it is blank. For nonpartitioned indexes on a partitioned table space, it is the logical partition number preceded by the character L (for example, L0001). For data-partitioned secondary indexes, the prefix for this value will be the character D (for example, D0001).

status Blank, or one or more of the following, delimited by commas. The maximum length of the string is 17 characters.

ACHKP
The table space is in aux check pending state. ACHKP on a table space indicates that CHECK DATA detected an orphan or missing LOB or an out-of-sync condition between the row in the base table and the LOB in the LOB table space.

ARBDP
Advisory Rebuild Pending status is set when indexes with varying length columns are changed from VARYING NO to VARYING YES.

AREO* The table space is placed in Advisory Reorg status. This may be because of a change in the table or index information. It also occurs because the table space, index, or partition identified in this status should be reorganized for optimal performance.

AREST The object (a table space, index space, or a physical partition of a table space or index space) is in an Advisory Restart Pending state. If backout activity against the object is not already underway, it should be initiated either by issuing the RECOVER POSTPONED command or by recycling the system with the system parameter LBACKOUT=AUTO.

AUXW The table space is in aux warning state. AUXW on a LOB table space indicates that one or more LOBs in the LOB table space is invalid. AUXW on a base table space indicates that CHECK DATA detected an orphan or missing LOB or an out-of-sync condition between the row in the base table and the LOB in the LOB table space.

CHKP The object (a table space or a partition within a table space) is in check pending status.

COPY The object (a table space or a partition within a table space) is in copy pending status. An image copy is required for this object.

GRECP
The object (a table space, a table space partition, index space, or index space partition) is marked recovery pending because of the failure of a a group buffer pool. This status value appears only on spaces. It does not apply to databases. See *DB2 Data Sharing: Planning and Administration* for information about resolving this restricted status.

ICOPY The object (an index space or index partition) is in informational copy pending status.

LPL The object has pages or ranges of pages that are unavailable because of logical or physical damage.

LSTOP The logical partition is stopped.

PSRBD
The index space is in page set rebuild pending status.

RBDP The object (an index space, index partition or logical index partition) is in rebuild pending status. For logical partitions, the RBDP status can appear as RBDP*.

RBDP Indicates that the entire index is inaccessible to SQL applications, however only the logical partition needs to be recovered.

RECP The object (a table space, table space partition, index space, index partition or logical index partition) is in recovery pending status.

REFP The object (a table space, a table space partition, index space, or index space partition) in in refresh pending status. For a table space, run LOAD REPLACE or the RECOVER utility to resolve this restricted status, or use the REBUILD INDEX utility to rebuild the index from the data.

REORP
The object (a table space or table space partition) is in recovery pending status.

RESTP The object is in the Restart Pending state. Backout activity against the object must be completed by using the -RECOVER POSTPONED command, or by recycling the system with system parameter LBACKOUT=AUTO.

RO The table space, table space partition, index space or index space partition is started for read-only activity.

RW The table space, table space partition, index space or index space partition is started for read and write activity.

STOP The table space, table space partition, index space or index space partition is stopped.

STOPE The table space or index space was implicitly stopped because there is a problem with the log RBA in a page. Message DSNT500I or DSNT501I is issued when the error is detected to indicate the inconsistency.

STOPP A stop is pending for the database, table space, table space partition, index space or index space partition.

UT The table space, table space partition, index space or index space partition is started for utility processing only.

UTRO A utility is in process on the table space, table space partition, index space, or index space partition that allows only RO access.

UTRW A utility is in process on the table space, table space partition, index space or index space partition that allows RW access.

UTUT A utility is in process on the table space, table space partition, index space or index space partition that allows only UT access.

WEPR This object has pages or ranges of pages that are unavailable because of physical damage. To reset this flag, you can use either the RECOVER utility with the option as ERROR RANGE or the LOAD utility with the option as REPLACE.

If the status for a nonpartitioned index is blank, then the status information is displayed at the logical partition level. If the type is TB, the status is blank.

conn-id A connection identifier for the thread.

corr-id A correlation identifier of the thread associated with the space name.

claims The type and duration of currently held claims. The claim is composed of a claim class descriptor and a claim duration descriptor.

The claim class descriptor has one of the following values:

CS Cursor stability claim class

RR Repeatable-read claim class

WR Write claim class

The claim duration descriptor describes the duration of the claims as follows:

A The claim is held until deallocation.

C The claim is held until the next commit point.

H The claim is held across commit.

DSNT390I *** DISPLAY OF DATABASE name action***

Explanation: This message indicates the end of a set of DSNT391I and DSNT392I messages, a set of DSNT393I and DSNT394I messages, or a set of DSNT395I and DSNT396I messages.

In this message:

- The 'name' is the name of the database.
- The 'action' is one of the following:

CONTINUES
The remaining table space and index space names within this database are continued to a new display.

TERMINATED
The remaining table space and index space names within this database are terminated.

ENDS The display for this database is complete.

DSNT391I **NAME TYPE STATUS PHYERRLOW PHYERRHI CATALOG PIECE**

Explanation: This message indicates the beginning of a set of messages used to display information as a result of a DISPLAY DATABASE command when the keywords WEPR, LOCKS, LPL, CLAIMERS and USE are omitted. (The message appears in one continuous line across the screen, not as shown above.) This message forms the headings of the display and is always followed by multiple DSNT392I messages containing the column data. Refer to message DSNT392I for a description of the data these headings identify.

DSNT392I *space type part status lo-page hi-page cat-name piece*

Explanation: This message contains the display information as a result of a DISPLAY DATABASE command when the keywords LOCK, CLAIMERS, LPL and USE are omitted. This message contains the column data describing the status of a table space or index space in the database listed in the preceding DSNT362I message. This message is always preceded by its column headings in message DSNT391I. The message number for this message is not displayed. However, messages DSNT391I, DSNT392I, and DSNT390I are always produced with a multiple line WTO and are never interrupted.

In this message:

- The *space* is the table space name or index space name.
- The *type* is TS for a table space, IX for an index space or LS for a LOB table space.
- The *part* is the partition number. It is blank for a simple table space, simple index space, or segmented table space. For nonpartitioned secondary indexes, it

is the logical partition number preceded by the character L (for example, L0001). For data-partitioned secondary indexes, the prefix for this value will be the character D (for example, D0001).

- See DSNT389I for a list of the possible *status* display strings.
- When a data set supporting the specified table space or index space has physical I/O errors, *lo-page* is a hexadecimal number identifying the lowest page number containing an I/O error. This column is blank if there are no physical errors.
- When a data set supporting the specified table space or index space has physical I/O errors, *hi-page* is a hexadecimal number identifying the highest page number containing an I/O error. This column is blank if there are no physical errors.
- When a data set supporting the specified table space or index space has physical I/O errors, *cat-name* identifies the catalog owning the data set with the I/O errors. This column is blank if there are no physical errors.
- When a data set supporting the specified table space or index space has physical I/O errors, *piece* is an integer number identifying the piece within the table space or index space having the error. This column is blank if there are no physical errors.

  The piece number can be used to construct the name of the data set supporting the space containing the physical error as follows:

  ```
  cat-name.DSNDBC.databas
  name.space-name.I0001.Axxxx
  ```

  The value xxxx is the piece number plus one and database-name is the name displayed in the preceding DSNT362I message. If the table space or index space is partitioned, the partition number should equal the piece number plus one.

DSNT393I NAME TYPE PART STATUS CONNID CORRID USERID

Explanation: This message indicates the beginning of a set of messages used to display information as a result of a DISPLAY DATABASE command when the keyword USE is specified. This message contains the headings of the display. (The message appears in one continuous line across the screen, not as shown above.) This message is always followed by multiple DSNT394I messages containing the column data. Refer to message DSNT394I for a description of the data these headings identify.

DSNT394I *space type part status conn-id corr-id user-id*

Explanation: This message contains the display information resulting from a DISPLAY DATABASE command when the keyword USE is specified. This message contains the column data describing the status of a table space or index space in the database listed in the preceding DSNT362I message. This message is always preceded by message DSNT393I containing the column headings. The message number for this message is not displayed. However, messages DSNT393I, DSNT394I, and DSNT390I are always produced with a multiple line WTO and are never interrupted.

In this message:

- The *type* is TS for a table space, IX for an index space or LS for a LOB table space.
- The *type* is TS for a table space or IX for an index space.
- The *part* is the partition number. It is blank for a simple table space, simple index space, or segmented table space. For nonpartitioned secondary indexes, it is the logical partition number preceded by the character L (for example, L0001). For data-partitioned secondary indexes, the prefix for this value will be the character D (for example, D0001).
- See DSNT389I for a list of the possible *status* display strings.
- The *conn-id* is a connection identifier for the thread. It is one of the following:
  - BATCH if the thread is from a batch region.
  - TSO if the thread is from a TSO terminal.
  - UTILITY if the thread is from a utility.
  - 'IMS identifier' if the thread is from an IMS terminal.
  - 'CICS identifier' if the thread is from a CICS terminal.
  - CONSOLE if the thread is from the console.
- The *corr-id* is a correlation identifier of the thread associated with the space name. It is one of the following:
  - 'job name' if the thread is from a batch region.
  - 'TSO logon identifier' if the thread is from a TSO terminal.
  - PST#.PSBNAME if the thread is from an IMS terminal.
  - 'entry ID.thread number.tran id' if the thread is from a CICS terminal.
  - CONSOLE if the thread is from the console.
  - '0nn', where 'nn' is a number , if the thread is for a system agent.
  - For distributed database access threads, this field contains the following:
    - While the thread is being created:
      1. ’028.DBAA nn’, where nn ranges from 01 to 99. This indicates that the database access agent (DBAA) is in its connection processing, which consists of establishing the DBAA thread, and validating and verifying the user’s ID. Refer to ″Controlling Access to a DB2 Subsystem″ in *DB2*

Administration Guide for descriptions of the authorization checks that might be performed.

All RACF requests are serialized on a single DB2 RACF service task, which might increase the validation time when multiple DBAAs are being created concurr

- After the thread is created:
  1. For threads using system-directed access and threads using application-direct access from a DB2 requester, this variable displays the correlation ID of the thread at the requesting location.
  2. For threads using application-directed access from a non-DB2 requester, this variable displays the first twelve characters in the DDM external name (EXTNAM) parameter of the DDM EXCSAT command received as part of the SQL CONNECT

- The *user-id* is one of the following:
  – 'TSO logon identifier' if the thread is from a TSO terminal.
  – 'sign-on id' or the LTERM if the thread is from an IMS message-drive region.
  – AXBUSER if the thread is from an IMS message-driven region and RACF is used.
  – 'PSB name' if the thread is from an IMS nonmessage-driven region and RACF is not used.
  – 'user-id', 'term id', 'tran id' or other, as specified in the resource control table (RCT), if the thread is from CICS.

DSNT395I NAME TYPE PART STATUS CONNID CORRID LOCKINFO

Explanation: This message indicates the beginning of a set of messages used to display information as a result of a DISPLAY DATABASE command when the keyword LOCKS is specified. This message forms the headings of the display. (The message appears in one continuous line across the screen, not as shown above.) This message is always followed by multiple DSNT396I messages containing the column data. Refer to message DSNT396I for a description of the data these headings identify.

DSNT396I *space type part status conn-id corr-id lock*

Explanation: This message contains the display information that results from a DISPLAY DATABASE command when the keyword LOCKS is specified. This message contains the column data describing the status of a table space or index space in the database listed in the preceding DSNT362I message. This message is always preceded by its column headings in message DSNT395I. The message number for this message is not displayed. However, the messages DSNT395I, DSNT396I, and DSNT390I are always produced with a multiple line WTO and are never interrupted.

In this message:

- The *space* is the table space name, index space name, or table OBID.
- The *type* is TS for a table space, IX for an index space, LS for a LOB table space or TB for a table.
- The *part* is the partition number. It is blank for a simple table space, simple index space, or segmented table space. For nonpartitioned secondary indexes, it is the logical partition number preceded by the character L (for example, L0001). For data-partitioned secondary indexes, the prefix for this value will be the character D (for example, D0001).
- See DSNT389I for a list of the possible *status* display strings. If the *type* is TB, the *status* field is blank.
- The *conn-id* is a connection identifier for the thread. It is one of the following:
  – BATCH if the thread is from a batch region.
  – TSO if the thread is from a TSO terminal.
  – UTILITY if the thread is from a utility.
  – 'IMS identifier' if the thread is from an IMS terminal.
  – 'CICS identifier' if the thread is from a CICS terminal.
  – CONSOLE if the thread is from the console.

  When the displayed lock is a P-lock or a retained lock, *conn-id* is not specified.
- The *corr-id* is a correlation identifier of the thread associated with the space name. It is one of the following:
  – 'job-name' if the thread is from a batch region.
  – 'TSO logon identifier' if the thread is from a TSO terminal.
  – PST#.PSBNAME if the thread is from an IMS terminal.
  – 'entry ID.thread number.tran id' if the thread is from a CICS terminal.
  – CONSOLE if the thread is from the console.
  – For distributed database access threads, this field contains the following:
    - While the thread is being created:
      1. '028.DBAA nn', where nn ranges from 01 to 99. This indicates that the database access agent (DBAA) is in its connection processing, which consists of establishing the DBAA thread, and validating and verifying the user's ID. Refer to "Controlling Access to a DB2 Subsystem" in *DB2 Administration Guide* for descriptions of the authorization checks that might be performed.

         All RACF requests are serialized on a single DB2 RACF service task, which might

increase the validation time when multiple DBAAs are being created concurr

- After the thread is created:
  1. For threads using system-directed access and threads using application-direc access from a DB2 requester, this variable displays the correlation ID of the thread at the requesting location.
  2. For threads using application-directed access from a non-DB2 requester, this variable displays the first twelve characters in the DDM external name (EXTNAM) parameter of the DDM EXCSAT command received as part of the SQL CONNECT

corr-id is not specified when the displayed lock is a P-lock or a retained lock.

- The *lock* describes the type and duration of locks:
  - Transaction locks (also called L-locks) are identified by their correspondin agents' connection and correlation IDs

    Multiple locks held by an agent are listed on multiple lines of the report followed by any locks the agent is currently awaiting.
  - Physical locks (P-locks) and retained locks are identified by their DB2 data sharing member names.

    Multiple P-locks or retained locks held by a DB2 data sharing member are listed on multiple lines of the report followed by any locks the member is currently awaiting.

  The *lock* is composed of a 'lock status' and followed by a dash '-', a 'lock state' followed by a 'lock type', and a 'lock duration' delimited by commas.
  - The 'lock status' is a value of:

    H The lock is held by the agent or the DB2 member.

    W The agent or the member is awaiting the lock.

    R The lock is retained.
  - The 'lock state' describes themode of the lock as follows:
    - For L-locks and drain locks,

      IS A lock with read intentions.

      IX A lock with update intentions.

      S A lock with read-only capability.

      U A lock with update capability.

      SIX A table space or table lock with a protocol that does not lock a page while reading, but locks the page with update intention while updating.

      X An exclusive lock.
    - For L-locks on LOB table spaces,

      IS A lock with intent to reserve space within the LOB table space

      IX A lock with intent to insert LOBs in the LOB table space.

      S A lock which reserves space within the LOB table space.

      SIX A lock which allows the owner to read and update LOBs.

      X An exclusive lock.
    - For P-locks,

      IS This DB2 member has R/O interest. The page set or partition is group buffer pool (GBP) dependent.

      IX This DB2 member has R/W interest. The page set or partition is GBP-depende

      S This DB2 member has R/O interest. The page set or partition is not GBP-dep

      SIX This DB2 member has R/W interest. The page set or partition is GBP-depende

      NSU This DB2 member has R/W interest. The page set or partition is GBP-depende

      X This DB2 member has R/W interest. The page set or partition is not GBP-dep
  - The 'lock type' describes the type of locking as follows:

    S A table space L-lock

    T A table L-lock

    C A cursor-stability read drain lock

    R A repeatable-read drain lock

    W A write drain lock

    P A partition L-lock

    PP A page set or partition P-lock
  - If the 'lock status' is H, the 'lock duration' describes the duration of the lock:

    A The lock is freed at deallocation.

    C The lock is freed at commit.

    H The lock is freed when all cursors are closed.

    M The lock is freed by the system.

    P The lock is freed when the plan is complete.

I The lock is a page set or partition P-lock. It is released when the pageset or partition data set is closed.

– If the 'lock status' is W, the 'lock duration' is a 2-digit number indicating the waiter's position in the local DB2 system's lock waiting queue. When this lock is acquired, its duration is the same as all locks held by the user.

– For a P-lock, the lock is owned by the system, not the user, and its duratio is not related to the duration of the locks that are owned by the user.

– Retained locks do not have a lock duration.

All index spaces on tables are implicitly locked with transaction locks at the same level as the table spaces containing those tables. P-locks are acquired by DB2 on index spaces as well as on table spaces.

DSNT397I

Explanation: This message indicates the beginning of a multiple line display generated as a result of a DISPLAY DATABASE command. The format of this multiple line display has three parts:

- The column headers, which are dependent on the keywords issued. The headers are underlined.
- The space information for all spaces and partitions requested by the command
- An ending message to state whether the spaces were displayed or not.

There are six different headers for the DISPLAY DATABASE commands. The headers are dependent on the presence of the keywords: CLAIMERS, LOCKS, USE, LPL, OVERVIEW, or no keyword. The five different DISPLAY DATABASE command formats, that have keywords, are described below.

- DISPLAY DATABASE CLAIMERS

  This DISPLAY DATABASE command displays the claims that are on certain table spaces, index spaces, and partitions.

  The column headers in this command are as follows:

  NAME TYPE PART STATUS CONNID CORRID CLAIMINFO

  – NAME is the table space name or index space name.

  – TYPE is TS for a table space, IX for an index space, LS for a LOB table space.

  – PART is the partition number. It is blank for a simple table space or simple index space. For nonpartitioned indexes on a partitioned table space, it is the logical partition number preceded by the character L (for example, L0001). For data-partitioned secondary indexes, the prefix for this value will be the character D (for example, D0001).

  – See DSNT389I for a list of the possible *status* display strings.

  – For L-locks on LOB table spaces,

  IS A lock with intent to reserve space within the LOB table space

  IX A lock with intent to insert LOBs in the LOB table space.

  S A lock which reserves space within the LOB table space.

  SIX A lock which allows the owner to read and update LOBs.

  X An exclusive lock.

  If the *type* is TB, the *status* field is blank.

  If the status for an index is blank, then the status information is displayed at the logical partition level.

  – The CONNID is a connection identifier for the thread. It is one of the following values:

  - BATCH if the thread is from a batch region.
  - TSO if the thread is from a TSO terminal.
  - UTILITY if the thread is from a utility.
  - 'IMS identifier' if the thread is from an IMS terminal.
  - 'CICS identifier' if the thread is from a CICS terminal.
  - CONSOLE if the thread is from the console.
  - For distributed database access threads, this field contains the following:
    - For threads using system-directed access and for threads using application-directed access from a DB2 Version 3 or later requester, this variable displays the connection name of the thread at the requesting location.
    - For threads using application-directed access from a non-DB2 requester or from a DB2 Version 2 Release 3 requester, this field displays the constant SERVER.

  – CORRID is a correlation identifier of the thread associated with the space name. It is one of the following values:

  - 'job name' if the thread is from a batch region.
  - 'TSO logon identifier' if the thread is from a TSO terminal.
  - PST#.PSBNAME if the thread is from an IMS terminal.
  - 'entry ID.thread number.tran id' if the thread is from a CICS terminal.
  - CONSOLE if the thread is from the console.
  - '0nn', where 'nn' is a number, if the thread is for a system agent.
  - For distributed database access threads, this field contains the following:
    - While the thread is being created:

1. '028.DBAA nn', where nn ranges from 01 to 99. This indicates that the database access agent (DBAA) is in its connection processing, which consists of establishing the DBAA thread, and validating and verifying the user's ID. Refer to "Controlling Access to a DB2 Subsystem" in *DB2 Administration Guide* for descriptions of the authorization checks that might be performed.

   All RACF requests are serialized on a single DB2 RACF service task, which might increase the validation time when multiple DBAAs are being created concurr

- After the thread is created:
  1. For threads using system-directed access and threads using application-direc access from a DB2 requester, this variable displays the correlation ID of the thread at the requesting location.
  2. For threads using application-directed access from a non-DB2 requester, this variable displays the first twelve characters in the DDM external name (EXTNAM) parameter of the DDM EXCSAT command received as part of the SQL CONNECT

– CLAIMINFO is the type and duration of currently held claims. The claim is composed of a claim class descriptor and a claim duration descriptor.

  The claim class descriptor has one of the following values:

  CS Cursor stability claim class
  RR Repeatable-read claim class
  WR Write claim class

  The claim duration descriptor describes the duration of the claims as follows:

  A The claim is held until deallocation.
  C The claim is held until the next commit point.
  H The claim is held across commit.

- DISPLAY DATABASE LOCKS

  This DISPLAY DATABASE command displays the locks that are on certain table spaces, index spaces and partitions.

  The column headers in this command are as follows:

  NAME TYPE PART STATUS CONNID CORRID LOCKINFO

  – The NAME column states the table space name, index space name, or table object ID.

  – The TYPE column states TS for table space, IX for index, or TB for table.

  – PART is the partition number. It is blank for a simple table space or simple index space. For nonpartitioned indexes on a partitioned table space, it is the logical partition number preceded by the character L (for example, L0001). For data-partitioned secondary indexes, the prefix for this value will be the character D (for example, D0001).

  – The STATUS, CONNID, and CORRID columns are described in the DISPLAY DATABASE CLAIMERS section.

  – The *conn-id* and *corr-id* are not specified when the display lock is a P-lock or a retained lock (either L-lock or P-lock). For page set P-locks, the DB2 member name is used to identify the lock.

  – The LOCKINFO describes the type and duration of locks used by corresponding agents identified by their connection and correlation IDs (for transaction locks, which are also called L-locks) or used by corresponding subsystems identified by their DB2 data sharing member names (for physical locks, which are called P-locks, and retained locks). Multiple locks held by an agent are listed on multiple lines of the report followed by any locks the agent is currently awaiting. Multiple P-locks or retained locks held by a DB2 data sharing member are listed on multiple lines of the report followed by any locks the member is currently awaiting. The LOCKINFO consists of a 'lock status', followed by a dash '-', a 'lock state', a 'lock type' and a 'lock duration' delimited by commas.

    - The 'lock status' is a value of:

      H The lock is held by the agent or the DB2 member.
      W The agent or the member is awaiting the lock.
      R The lock is retained.

    - The 'lock state' describes the mode of lock as follows:

      - For L-locks and drain locks,

        IS A lock with read intentions.
        IX A lock with update intentions.
        S A lock with read-only capability.
        U A lock with update capability.
        SIX A lock with a protocol that does not lock a page while reading, but locks the page with update intention while updating.
        X An exclusive lock.

      - For P-locks,

        IS This DB2 member has R/O interest. The page set or partition is group buffer pool (GBP) dependent.

        IX This DB2 member has R/W interest. The page set or partition is GBP-depend

        S This DB2 member has R/O interest. The page set or partition is not GBP-dependent.

SIX This DB2 member has R/W interest. The page set or partition is GBP-depend

NSU This DB2 member has R/W interest. The page set or partition is GBP-depend

X This DB2 member has R/W interest. The page set or partition is not GBP-dependent.

- The 'lock type' describes the type of locking as follows:

  S A table space L-lock

  T A table L-lock

  C A cursor-stability read drain lock

  R A repeatable-read drain lock

  W A write drain lock

  P A partition L-lock

  PP A page set or partition P-lock

- If the 'lock status' is H,the 'lock duration' describes the duration of the lock:

  A The lock is freed at deallocation.

  C The lock is freed at commit.

  H The lock is freed when all cursors are closed.

  M The lock is freed by the system.

  P The lock is freed when the plan is complete.

  I The lock is a page set or partition P-lock. It is released when the page set or partition data set is closed.

- If the 'lock status' is W, the 'lock duration' is a 2-digitnumber indicating the waiter's position in the local DB2 system's lock waiting queue. When this lock is acquired, its duration is the same as all locks held by the user.
- For a P-lock, the lock is owned by the system, not the user, and its duration is not related to the duration of the locks that are owned by the user.
- Retained locks do not have a lock duration.

There are no explicit index space locks. However, all index spaces on tables are implicitly locked at the same level as the table spaces containing those tables. P-locks are acquired by DB2 on index spaces as well as on table spaces.

- DISPLAY DATABASE USE

  This DISPLAY DATABASE command displays the user IDs that are accessing certain table spaces, index spaces, and partitions.

  The column headers in this command are as follows:

  NAME TYPE PART STATUS CONNID CORRID USERID

  – NAME is the table space name or index space name.
  – TYPE is TS for a table space, IX for an index space, LS for a LOB table space.
  – PART is the partition number. It is blank for a simple table space or simple index space. For nonpartitioned indexes on a partitioned table space, it is the logical partition number preceded by the character L (for example, L0001). For data-partitioned secondary indexes, the prefix for this value will be the character D (for example, D0001).
  – The STATUS, CONNID, and CORRID columns are described in the DISPLAY DATABASE CLAIMERS section.
  – The USERID is one of the following variables:
    - 'TSO logon identifier' if the thread is from a TSO terminal.
    - 'sign-on id' or the LTERM if the thread is from an IMS message-drive region.
    - AXBUSER if the thread is from an IMS message-driven region and RACF is used.
    - 'PSB name' if the thread is from an IMS nonmessage-driven region and RACF is not used.
    - 'userid', 'term id', 'tran id' or other, as specified in the resource control table (RCT), if the thread is from CICS.

- DISPLAY DATABASE LPL

  NAME TYPE PART STATUS LPL PAGES

  This message contains the display information resulting from a DISPLAY DATABASE command when the keyword LPL is specified. It contains the column data describin the status of a table space or index space in the database listed in the preceding DSNT362I message. This message is always preceded by its column headings message DSNT805I. The message number for this message is not displayed.

  In this message:

  – The LPL PAGES column lists the pages in the specified *space* that are unavailable because of logical or physical damage.

- DISPLAY DATABASE (with no LOCKS, USE, LPL or CLAIMERS keyword).

  This DISPLAY DATABASE command displays the error page range information for certain table spaces, index spaces, and partitions.

  The column headers in this command are as follows:

  NAME TYPE PART STATUS PHYERRLO PHYERRHI CATALOG PIECE

  In this message:

  – NAME is the table space name or index space name.

- TYPE is TS for a table space, IX for an index space, LS for a LOB table space.
- PART is the partition number. It is blank for a simple table space or simple index space. For nonpartitioned indexes on a partitioned table space, it is the logical partition number preceded by the character L (for example, L0001). For data-partitioned secondary indexes, the prefix for this value will be the character D (for example, D0001).
- The STATUS columns are described in the DISPLAY DATABASE CLAIMERS section.
- When a data set supporting the specified table space or index space has physical I/O errors, PHYERRLO is a hexadecimal number identifying the lowest page number containing an I/O error. This column is blank if there are no physical errors.
- When a data set supporting the specified table space or index space has physical I/O errors, PHYERRHI is a hexadecimal number identifying the highest page number containing an I/O error. This column is blank if there are no physical errors.
- When a data set supporting the specified table space or index space has physical I/O errors, CATALOG identifies the catalog owning the data set with the I/O errors. This column is blank if there are no physical errors.
- When a data set supporting the specified table space or index space has physical I/O errors, PIECE is an integer number identifying the piece within the table space or index space having the error. This column is blank if there are no physical errors.

  The piece number can be used to construct the name of the data set supporting the space containing the physical error as follows:

  ```
  cat-name.DSNDBC.database-
  name.space-name.I0001.Axxxx
  ```

  The value *xxxx* is the piece number plus one and database-name is the name displayed in the preceding DSNT362I message. Note that if the table space or index space is partitioned, the partiti number should equal the piece number plus one.

There are other informational messages that can be displayed that are not related to one specific keyword. They are listed below:

- –luw-id ACCESSING DATA FOR location

  This message is issued for each thread reported that is participating in distributed data access (database access thread). The thread that corresponds to this message is the thread named by the correlation id and connection id of the data line preceding this message. This additional line reports the distributed identification as follows:

  luw-id=token
  The logical-unit-of-work identifier (LUW-ID) is a unique identifier that is assigned to the thread followed by an '=' sign and a token. The token can be used in place of luw-id in any DB2 command that accepts luw-id as input.

  location
  The LOCATION name assigned to the requesting site.

- –MEMBER NAME *member-name*

  This message is issued with messages DSNT389I, DSNT394I or DSNT396I for each thread in a data sharing environment. This additional line reports the member name of the DB2 subsystem accessing this object.

  If the characters '(CO)' follow the member name, then this member is the castout owner for the page set or partition. The castout owner for a page set or partition is the DB2 member responsible for doing all of the castout I/O for that page set or partition.

  The message number and subsystem recognition character are not included in this message.

- INVALID space-name

  The space-name specified in the SPACENAM keyword on the -DISPLAY DATABASE command does not exist. The name of the table space or index space is 'space-name'. The request is rejected.

- NO SPACES FOUND

  This message is produced when a DISPLAY DB(name) SPACENAM is entered and no spaces were found to satisfy the request. SPACENAM(...) might have been entered as a range, a subset, or a list of spaces.

- spacetype name NOT type

  This message is produced when no table space or index space listed in the DISPLAY DB(...) SPACENAM(...) satisfies your request.

  spacetype
  TABLESPACE if the space is a table space or INDEXSPACE if the space is an index space.

  name The name of the space.

  type A character string that indicates the type of request specified. The string is a list of the following, delimited by 'OR':
  - ACTIVE if the keyword ACTIVE is specified
  - RESTRICTED if the keyword RESTRICT is specified.

- NO INFORMATION AVAILABLE

  This message is produced when there is no LOCK or USE information available. When DISPLAY DB(...) is

entered, and DB(...) is a list of databases or a list of names, this message indicates that no listed database has LOCK or USE information available.

If DISPLAY DB(...) or DISPLAY DB(...) SPACENAM(*) is entered, and DB(...) is a range or subset of databases, or all databases ('*'), this message indicates that no databases, for which the invoker has authority to display, have LOCK or USE information available. This message is produced only when the keywords USE and LOCKS are specified.

The ending message looks like this:

* DISPLAY OF DATABASE name action *

In this message:

name The name of the database

action One of the following:

CONTINUES
The remaining table space and index space names within this database are continued to a new display.

TERMINATED
The remaining table space and index space names within this database are terminated.

ENDS The display for this database is complete.

DSNT398I

Explanation: This message appears within a multiple line WTO message set and precedes DSNT391I, DSNT393I, and DSNT395I. It is used to underscore the column headings.

This message is issued by CSECT: DSNTDMUL

DSNT399I INVALID space-name

Explanation: The space-name specified in the SPACENAM keyword on the -DISPLAY DATABASE command does not exist. The name of the table space or index space is 'space-name'. The request is rejected.

DSNT400I SQLCODE = 0, SUCCESSFUL EXECUTION

Explanation: The SQLCODE was 0, indicating successful execution of the function. SQLWARN indicators may note some warnings. SQLWARNx fields are described in Appendix D of *DB2 SQL Reference*.

DSNT401I DSNT401I WARNING - HELP IS NOT INSTALLED

Explanation: You invoked the installation process with the DSNTINST CLIST. To enable online help, you must use the DSNTINS0 CLIST instead of the DSNTINST CLIST.

This message is issued by panel: DSNTIPA1

System action: The panel is displayed with the warning message.

User response: No response is necessary if you do not desire online help for the panel. Press ENTER to continue with the installation process after filling in desired panel values. If panel help is desired: exit this panel, make sure BookManager or the online help reader is installed on your system, and invoke the installation process with the DSNTINS0 CLIST.

DSNT402I ONLINE BOOK DATA SET NAMES MUST END IN '.BOOK'

Explanation: All book names entered on this panel must end in ".BOOK".

This message is issued by panel: DSNTIPA0

System action: The panel is redisplayed with the error message.

User response: Replace the book name or names with a value that ends in ".BOOK".

DSNT404I SQLCODE = xxx, explanation

Explanation: This message contains an SQL return code and a brief explanation, with text inserted from the SQLERRM field. The SQL return code is positive, indicating a warning condition. For more information about this SQL return code, see *DB2 Codes*.

System programmer response: Determine the cause for the SQL warning code, and correct the situation if needed. Rerun the application program or SQL statement.

DSNT405E csect-name DISPATCH PRIORITIES NOT IN SYNC: ccccccc1: hhh1 COMPARED TO ccccccc2: hhh2

Explanation: The dispatch priorities of task 'ccccccc1' (dispatch priority 'hhh1') and task 'ccccccc2' (dipatch priority 'hhh2') are found to be incorrect in relation to each other. This can cause severe performance problems and possible failure of either task 'ccccccc1', task 'ccccccc2', or other tasks.

System action: DB2 continues with possible problems described above.

Operator response: Notify the system programmer.

System programmer response: Modify the dispatch priority algorithms for the two tasks to meet documented requirements. For instance, IRLM must have a higher dispatch priority than DB2.

DSNT406I THE ART/ORT ESCAPE CHARACTER CANNOT BE % OR _

Explanation: You specified the percent sign (%) or the underscore character (_) as the application registration table/object registration table (ART/ORT) escape character. The ART/ORT escape character can be any other character.

System action: The panel is redisplayed and the cursor is placed on the field that produced the error.

User response: Change the ART/ORT escape character value.

DSNT407I THIS FIELD HAS AN INVALID FORMAT

Explanation: The field must be NOLIMIT, 1440, 0, or minutes and seconds separated by a comma.

This message is issued by the following panel: DSNTIPA

System action: The panel is redisplayed and the cursor is placed on the field in error.

User response: Correct the input.

DSNT408I SQLCODE = -xxx, explanation

Explanation: This message contains an SQL return code and a brief explanation, with text inserted from the SQLERRM field. The SQL return code is negative, indicating that an error has occurred. For more information about this SQL return code, see *DB2 Codes*.

System programmer response: Determine the cause for the SQL error by using information about that specific SQLCODE. Correct the error if desired. Rerun the application program or SQL statement.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 5, 14, 17.

DSNT409I BUFFERPOOL SIZES SHOWN MAY BE OBSOLETE

Explanation: The buffer pool sizes shown here were set at installation or migration time and might not be the most current. Updates might have occurred using the ALTER BUFFERPOOL command.

System action: Processing continues.

User response: No response is required.

DSNT410I THE SUBSYSTEM COMMAND PREFIX ENTERED IS INCORRECT

Explanation: You entered an incorrect command prefix. The first character of the command prefix must be one of the following:

@ $ # . / ' ? " % ! <
) * + – = ¢ : _ ; |

The remaining characters of the command prefix must be one of the above, A-Z, or 0-9.

This message is issued by the following panel: DSNTIPM

System action: The panel is redisplayed and the cursor is placed on the field in error

User response: Enter a valid command prefix

DSNT411I SQLCAID = xxxx, SQLCA IDENTIFIER

Explanation: The SQLCA ID field (SQLCAID) is not the expected value, C'SQLCA'.

System programmer response: The SQLCA area is invalid. This type of error is usually the result of one of the following:

- Calling the DSNTIAR formatting routine before an SQL statement initializes the area
- Overlaying the SQLCA area
- Passing the wrong address to the DSNTIAR routine.

Correct the error, and rerun the application program.

DSNT412I SQLCABC = xxxx, SQLCA BYTE COUNT

Explanation: The size of the SQLCA is not the expected value (136 bytes).

System programmer response: The SQLCA area is invalid. Correct the error. It is probably an overlay or an incorrect SQLCA address. Rerun the application program.

DSNT413I SQLCODE = xxxx, INVALID SQLCODE

Explanation: The SQL return code is not a valid value.

System programmer response: Either the SQLCA area is invalid or the maintenance was not coordinated correctly. As a result, the SQL return code was added to the DB2 subsystem but not to the formatting routine.

An invalid SQLCA area is usually the result of one of the following:

- Calling the DSNTIAR formatting routine before an SQL statement initializes the area
- Overlaying the SQLCA area
- Passing the wrong address to the DSNTIAR routine.

If the maintenance was not coordinated correctly, the DSNTIAR and DSNTIAM CSECTs were probably not replaced in the code that is now running. Link-edit the load module, and replace DSNTIAR and DSNTIAM. Correct the error, and rerun the application program.

DSNT414I SQLERRM = xxxx, SQLCA ERROR MESSAGE INSERTION TEXT

Explanation: This message results from an SQL return code that is not recognized by DSNTIAR. SQLERRM contains error message text.

System programmer response: Either the SQLCA area is invalid or the maintenance was not coordinated correctly. As a result, the SQL return code was added to the DB2 subsystem but not to the formatting routine.

An invalid SQLCA area is usually the result of one of the following:

- Calling the DSNTIAR formatting routine before an SQL statement initializes the area
- Overlaying the SQLCA area
- Passing the wrong address to the DSNTIAR routine.

If the maintenance was not coordinated correctly, the DSNTIAR and DSNTIAM CSECTs were probably not replaced in the code that is now running. Link-edit the load module, and replace DSNTIAR and DSNTIAM. Correct the error, and rerun the application program.

DSNT415I SQLERRP = xxxx, SQL PROCEDURE DETECTING ERROR

Explanation: The name of the procedure detecting the error is provided in the SQLERRP.

System programmer response: (The SQL procedure detecting the error may be of help in diagnosing an error or warning.) Correct the error if there is one noted in other messages. Rerun the program or SQL statement.

DSNT416I SQLERRD = *xx,xx,xx,xx,xx,xx* SQL DIAGNOSTIC INFORMATION

Explanation: One or more of the six diagnostic fields do not have the expected value. Check that all of the diagnostic fields contain one of the following expected values:

- 00, 00, 00, 00, 00, 00 or
- 00, 00, 00, -1, 00, 00

The SQL diagnostic information for each of the six values is as follows:

1 DB2 unique internal relational data system code

2 DB2 internal data manager error code

3 One of the following conditions:

- Number of rows inserted, updated, or deleted following an INSERT, UPDATE, or DELETE statement. **Attention:** A '-1' in this field indicates that a mass delete of a table in a segmented table space was performed, wherein DB2 does not access nor count individual rows.
- For SQLCODE -911 or -913, a reason code that indicates whether a deadlock or timeout occurred. **Tip:** The most common reason codes are:
  - 00C90088 - deadlock
  - 00C9009E - timeout

4 Reserved

5 Position or column of a syntax error for PREPARE or EXECUTE IMMEDIATE

6 DB2 internal buffer manager error code

System programmer response: Refer to Appendix D of *DB2 SQL Reference* for further information on the contents of the SQLERRD fields. The SQL diagnostic information might help in diagnosing an error or in indicating the results of successful execution. If this error or warning should not occur, correct the error. Rerun the program or SQL statement.

DSNT417I SQLWARNn-n SQL WARNINGS

Explanation: At least one of the 11 warning values is not the expected value (a blank). A nonblank value in one of the 11 warning fields has the following meaning:

SQLWARN0
Any other warning code is set.

SQLWARN1
String truncation.

SQLWARN2
Null value(s) eliminated.

SQLWARN3
Number of host variables provided did not equal degree of the table.

SQLWARN4
No WHERE clause on UPDATE or DELETE.

SQLWARN5
Not a DB2 SQL statement.

SQLWARN6
Date or timestamp value adjusted.

SQLWARN7
Conversion of decimal number from precision 31 to precision 15 caused truncation of fractional digits.

SQLWARN8
Untranslatable character replaced with substitute.

SQLWARN9
Arithmetic exceptions have been ignored during processing of COUNT DISTINCT.

SQLWARNA
At least one character field of the SQLCA is invalid due to a string translation error.

User response: The SQL warning information may be of help in diagnosing an error or in indicating the results of successful execution. If this warning should

occur, correct the error and rerun the program or SQL statement.

DSNT418I SQLSTATE = sqlstate, SQLSTATE RETURN CODE

Explanation: The SQLSTATE is a return code that indicates the outcome of the most recently executed SQL statement. The execution of every SQL statement sets SQLSTATE to a five-digit code in the range of 00000 to 65535. This has no effect on the existing use of any other field in the SQLCA.

User response: For further information on the meaning of an SQLSTATE return code, see the *DB2 Codes*. If necessary, correct the SQL statement and return it.

DSNT419I WARNING - MAX REMOTE ACTIVE PREVIOUSLY SPECIFIED AS 0.

Explanation: You specified a value of 0 for the number of active database access threads on a previous panel. This value might be inconsistent with a DDF STARTUP OPTION of AUTO or COMMAND.

This message is issued by the following panel: DSNTIPR

Severity: 4 (warning)

System action: The panel is redisplayed and the cursor is placed on the DDF STARTUP OPTION field.

User response: No response is necessary if your intent is to have DDF started with no active database access threads allocated and DDF is used as a requester only. Press ENTER to continue. Otherwise, either change the DDF STARTUP OPTION to NO or MAX REMOTE ACTIVE to the number of threads you want.

DSNT420I YOU NEED AT LEAST 1 32K DS IF THE TEMP 32K SPACE IS GREATER THAN 0

Explanation: You entered zero for the TEMP 32K DATA SETS field on the SIZES panel. You need at least one TEMP 32K DATA SET if TEMP 32K SPACE on DSNTIPD is not zero.

This message is issued by the following panel: DSNTIPD

System action: The panel is redisplayed and the cursor is placed on the field in error.

User response: Enter a value of one or greater for the TEMP 32K DATA SETS field or change the TEMP 32K SPACE value to zero.

DSNT421I YOU NEED AT LEAST 1 32K BUFFER FOR THE <*num32kds*> 32K DATA SETS

Explanation: You specified zero for the 32KB buffers. If field TEMP 32K DATA SETS on installation panel DSNTIPD is greater than zero, you need at least one 32KB buffer on installation panel DSNTIP2.

This message is issued by the following panel: DSNTIP2

System action: The panel is redisplayed and the cursor is placed on the field in error.

User response: Enter a value of 1 or more for at least one of the buffer pools (BP32K - BP32K9) on panel DSNTIP2, or change the TEMP 32K DATA SETS value to zero.

DSNT422I THE ART TABLE NAME MUST BE DIFFERENT FROM THE CRT TABLE NAME

Explanation: You entered the same name for the Application Registration Table and the Object Registration Table on the Data Definition Control Support panel.

This message is issued by the following panel: DSNTIPZ

System action: The panel is redisplayed and the cursor is placed on the field in error.

User response: Change either the APPL REGISTRATION TABLE field or the OBJT REGISTRATION TABLE field to make these two fields unique with respect to each other.

DSNT424I OUTPUT DATA SETS MUST BE DIFFERENT FROM THE INPUT DATA SETS

Explanation: You specified an output data set name that is the same as one of the input data set names built from the PREFIX and SUFFIX fields on the main panel DSNTIPA1.

This message is issued by the following panel: DSNTIPT

System action: The panel is redisplayed and the cursor is placed on the field in error.

User response: Either change the data set name under the cursor or change the PREFIX or SUFFIX fields on panel DSNTIPA1.

DSNT425I ALL THREE OUTPUT DATA SETS MUST BE UNIQUE

Explanation: You specified the same name for two or more of the output data sets.

This message is issued by the following panel: DSNTIPV

System action: The panel is redisplayed and the cursor is placed on the field in error.

User response: Change the data set name under the cursor or the data set field(s) that are the same as the one under the cursor.

DSNT426I ALL THREE DATA SETS ARE REQUIRED FOR OUTPUT

Explanation: You specified one or more output data sets and left one or more output data set fields blank.

This message is issued by the following panel: DSNTIPT

System action: The panel is redisplayed and the cursor is placed on the field in error.

User response: You must specify all or none of the output data set fields. Leave blank all output data set fields if you do not want the installation CLIST to edit any CLISTs or JCL.

DSNT427I THE DATA SET OR MEMBER SPECIFIED COULD NOT BE LOCATED

Explanation: The input parameter member for the INPUT MEMBER NAME field or the DATA SET NAME(MEMBER) field for migration could not be found.

This message is issued by the following panel: DSNTIPA1

System action: The panel is redisplayed and the cursor is placed on the field in error.

User response: Verify that you entered an existing data set and member in that data set. Also verify that the data set is cataloged.

DSNT428I WHEN USING TIMESTAMPS, THE PREFIX MUST BE <= *preflen* CHARACTERS LONG

Explanation: YOU HAVE ENTERED AN ARCHIVE LOG PREFIX LONGER THAN ALLOWED. The archive log prefix names must not exceed 19 characters when the TIMESTAMP ARCHIVES value is YES and must not exceed 17 characters when the TIMESTAMP ARCHIVES value is EXT.

This message is issued by the following panel: DSNTIPH

System action: The panel is redisplayed and the cursor is placed on the field in error.

User response: Reduce the length of the prefix or change the TIMESTAMP ARCHIVES value to NO.

 # DSNT429I THE SUM OF MAX USERS AND MAX # REMOTE ACTIVE MUST BE <= TO # 2000

Explanation: The total number of threads determined by MAX USERS and MAX REMOTE fields is too large.

System action: The panel is redisplayed and the cursor is placed on the field in error.

User response: Reduce the MAX USERS field, or the MAX REMOTE field, or both.

DSNT430I THIS FIELD MAY ONLY BE FILLED IN FOR MIGRATE MODE

Explanation: You can only specify the FROM RELEASE or DATA SET NAME(MEMBER) fields for INSTALL TYPE of MIGRATE.

This message is issued by the following panel: DSNTIPA1

System action: The panel is redisplayed and the cursor is placed on the field in error.

User response: Change the INSTALL TYPE to MIGRATE or blank out the field under the cursor.

DSNT431I DDF, MIXED DATA, OR DEF ENCODING SCHEME CONFLICTS WITH CODED CHAR SET

Explanation: You have entered an invalid EBCDIC or ASCII coded character set (CCSID) value for the DDF, MIXED DATA, or DEF ENCODING SCHEME value you specified. The following restrictions apply to these values:

- With a MIXED DATA value of 'YES', you must choose a valid mixed CCSID value for the EBCDIC CODED CHAR SET. The ASCII CODED CHAR SET must be a valid mixed value or zero.
- With a MIXED DATA value of 'NO', you must choose a valid SBCS CCSID value or zero for EBCDIC or ASCII CODED CHAR SETs.
- With a DDF value of 'AUTO' or 'COMMAND', you must choose a non-zero value for the EBCDIC CODED CHAR SET.
- With a DEF ENCODING SCHEME value of 'ASCII', you must choose non-zero values for EBCDIC and ASCII CODED CHAR SETs.

Refer to Appendix C of *DB2 Installation Guide* for a list of the best general CCSID for each country.

This message is issued by the following panels: DSNTIPF, DSNTIPR

System action: The panel is redisplayed and the cursor is placed on the field in error.

User response: Correct either the DDF, MIXED DATA, or DEF ENCODING SCHEME value or the EBCDIC or

ASCII CODED CHAR SET value to correct the incompatibility.

DSNT433I MAX REMOTE ACTIVE MUST BE <= MAX REMOTE CONNECTED.

Explanation: The number of active database access threads cannot be larger than the number of concurrent database access threads.

This message is issued by the following panel: DSNTIPE

Severity: 8 (error)

System action: The panel is redisplayed and the cursor is placed on the field in error.

User response: Reduce the MAX REMOTE ACTIVE field, increase the MAX REMOTE CONNECTED field, or both.

DSNT434I WARNING, DATA SETS MARKED WITH ASTERISKS EXIST AND WILL BE OVERWRITTEN

Explanation: The data sets marked with asterisks already exist. If you continue, the installation process will delete the existing data set and allocate a new one for tailored job output.

This message is issued by the following panel: DSNTIPT

Severity: 4

User response: You should:

1. Press ENTER to continue; the data set(s) will be deleted and reallocated OR
2. Press RETURN to leave the installation process; the data set(s) will not be deleted OR
3. Change the data set name(s); the new name will be checked for existence.

DSNT435I DSNT435I OVERRIDE VALUES ARE IN EFFECT

Explanation: Calculated values for EDMPOOL, SORTPOOL, or RIDPOOL on panel DSNTIPC were previously overridden. Installation uses any override values displayed.

Severity: 4

System action: The display shows the override values instead of the calculated values.

User response: No response is necessary if you want to use the override values. If you want to use the calculated values, erase the values in the override column.

DSNT436I REGION SIZE MAY BE TOO LARGE

Explanation: The numbers you specified produced a region size estimate of more than 5000KB. This region size estimate does not include the storage required for SWA because it is assumed that SWA has been placed above the line. If a larger private area is permitted, you should increase the region size on the DB2 start procedures. The region size calculation is shown in the DSNT485I through DSNT487I messages and is discussed in Part 2 of *DB2 Installation Guide*.

This message is issued by the following installation CLIST: DSNTINST

Severity: 4 (warning)

System action: The CLIST continues.

User response: Check the values entered for numbers of concurrent users and databases, buffers, and plan sizes. Check the available private region size. If this region size is not permitted, these numbers need to be reduced. If the size is permitted, this warning can be ignored.

Problem determination: If you suspect an error in the process, set CONTROL(CONLIST) and invoke the CLIST. This produces tracing for error documentation.

DSNT437I REGION SIZE IS TOO LARGE

Explanation: The numbers you specified produced a region size estimate of more than 8000KB. This estimate does not include the storage required for SWA, because it is assumed that you placed SWA above the 16MB line. The region size calculation is discussed in Part 2 of *DB2 Installation Guide*.

This message is issued by the following installation CLIST: DSNTINST

Severity: 8 (error)

System action: The CLIST continues, but does not produce edited jobs.

User response: Check the numbers entered for numbers of concurrent users and databases, buffers, and plan sizes. One or more of these numbers needs to be reduced.

Problem determination: If you suspect an error in the process, set CONTROL(CONLIST) and invoke the CLIST. This produces tracing for error documentation.

DSNT438I WARNING, IRLM LOCK MAXIMUM SPACE = *irlmreg* G, AVAILABLE SPACE = *irlmav* G

Explanation: You specified a maximum number of locks per user that could use a total space larger than that available to the IRLM, causing it to abend. It is not likely that all threads will have the maximum number of locks, but if the maximum exceeds the amount of

space by a significant factor, the possibility of abends in the IRLM is increased.

The available space is specified in the MAX STORAGE FOR LOCKS field of the DSNTIPJ installation panel, and is never less than 2 GB. The maximum space used is estimated as the sum of the MAX USERS and MAX REMOTE ACTIVE values from the DSNTIPE panel times the number of LOCKS PER USER value on the DSNTIPJ panel times 540 bytes per lock.

Severity: 4 (warning)

System action: The CLIST will continue.

User response: Choose one of the following responses:
- Ensure that you can run with the overcommitment of lock address space, or
- Lower the value(s) of the MAX USERS, MAX REMOTE ACTIVE, or LOCKS PER USER fields, or
- Increase the value of the MAX STORAGE FOR LOCKS field.

DSNT439I IRLM NO LOCK ESCALATION OR NO USER MAXIMUM

Explanation: You specified the number 0 for:
- The number of table or table space locks that will cause lock escalation, or
- The number of locks that will terminate a DB2 user.

A lock escalation value of 0 turns off lock escalation and means that DB2 cannot use this means of conserving lock space. A user maximum of 0 means that the calculation noted in the DSNT438 message cannot be performed to see if the IRLM has sufficient address space for all users, since there is no maximum per user.

This message is issued by the following installation CLIST: DSNTINST

Severity: 4 (warning)

System action: The CLIST will continue. This is only a warning.

User response: Turn off lock escalation and/or user lock limiting.

DSNT440I TOTAL BP SIZE TOO LARGE

Explanation: The total size of a buffer pool is too large. The summation of the *ssnm*DBM1 address space storage requirements for all buffer pools cannot exceed 1TB. When calculating the storage requirements for the buffer pools, DB2 uses the page size (4K, 8K, 16K, or 32K) for each buffer.

System action: The panel is redisplayed and the cursor is placed on the field in error.

User response: Reduce one or more of the buffer pool sizes.

DSNT442E ERROR, VIRTUAL STORAGE OF main EXCEEDS THE SYSTEM LIMIT OF 2 GIGABYTES

Explanation: The total of all virtual storage calculations exceeds the maximum amount possible for an address space.

This message is issued by the following CLIST: DSNTINST

Severity: 8 (error)

System action: The installation CLIST returns to the initial panel without editing the installation jobs.

User response: Respecify installation values that determine storage sizes so that the 2 GB limit is not exceeded.

DSNT443I VALUES MARKED WITH AN ASTERISK HAVE BEEN UPDATED

Explanation: An asterisk is placed before the description of fields which were generated by the DSNTINST CLIST. This message only appears when you update field(s) from a prior panel used to generate the fields on the current panel.

This message is issued by the following panels: DSNTIPA, DSNITPL

System action: The panel is displayed.

User response: Verify that the current value is what you want. If it is not, enter the desired value.

DSNT444I SCROLLING BACKWARD MAY CHANGE FIELDS MARKED WITH ASTERISKS

Explanation: The marked fields on the current panel are generated from fields on other panels. If you scroll backward, these fields may be updated to reflect new values from other panels. For example, the ARCHIVE PREFIX field is generated from the CATALOG ALIAS field on DSNTIPA2. Message DSNT443I will be displayed to inform you when fields are changed by the DSNTINST CLIST.

This message is issued by the following panels: DSNTIPA, DSNITPL

System action: The panel is displayed.

User response: No action required.

DSNT445I REMOTE LOCATION NAME MUST NOT BE THE SAME AS LOCAL LOCATION

Explanation: The name entered for the remote sample location was the same as the local location name.

This message is issued by the following installation panel: DSNTIPY

System action: The panel is displayed.

User response: Enter a remote location name or leave the field blank to indicate that the organization sample will only be executed locally.

DSNT446I SAMPLE LOCATION MUST BE BLANK IF LOCAL LOCATION IS BLANK

Explanation: In order to edit the remote sample jobs, both the DB2 LOCATION NAME (local location) on panel DSNTIPR and the REMOTE LOCATION must have valid names. If the DB2 LOCATION NAME is unknown, leave both fields blank and the remote sample jobs will not be edited.

This message is issued by the following panel: DSNTIPY

System action: The panel is displayed.

User response: Either scroll back to the DISTRIBUTED DATA FACILITY panel and enter a value for the DB2 LOCATION NAME, or enter a blank in the REMOTE LOCATION field on the job editing panel.

DSNT448I STG AVAILABLE ABOVE 2GB = size T

Explanation: This is the estimated amount of real storage recommended to support the virtual storage sizes chosen.

The amount of real storage needed will vary greatly. Some customers will find that they need several times the estimate shown. For more information on storage calculations, refer to Part 2 of *DB2 Installation Guide*.

This message is issued by the following CLIST: DSNTINST

Severity: 0 (successful execution)

System action: The CLIST continues.

User response: Compare this recommended value with the actual available real storage.

DSNT449I NUMBER OF LOGS REDUCED FROM old-num TO MAXIMUM OF 31

Explanation: The number of active log data sets used in the DB2 release you are migrating from is greater than the new maximum number of data sets allowed for the current release of DB2. Installation job DSNTIJUZ will update your bootstrap data sets to contain only the last 31 data sets.

This message is issued by the following panel: DSNITPL

System action: The value is reduced to the maximum of 31 and the panel is displayed.

User response: Update any other fields on the panel and continue.

DSNT450I type ENTRY MUST BE EXACTLY TWO CHARACTERS

Explanation: These entries are used to complete SYS1.PARMLIB member names and must be exactly two characters. The type is:

APF AUTH
For the APF-authorized library list IEAAPFxx

LINK LIST
For the link list library concatenation LNKLSTxx

MVS SUBSYS
For the MVS subsystem list IEFSSNxx

This message is issued by the following installation CLIST: DSNTINST

Severity: 8 (error)

System action: The CLIST will continue.

User response: Check your entries for these fields on the MVS panel and make them exactly two characters. See job DSNTIJMV that was produced and correct the IEBUPDTE CHANGE statement.

DSNT451I WARNING-THE LOCAL DATE LENGTH OR LOCAL TIME LENGTH HAS BEEN INCREASED

Explanation: The LOCAL DATE LENGTH or LOCAL TIME LENGTH has been increased and a REBIND command must be issued for any dependent plans.

This message is issued by the following installation CLIST: DSNTINST

Severity: 4 (warning)

System action: Processing continues.

User response: Rebind all application plans that depend upon the length of the local format that was changed.

DSNT454I ERROR CODE = errcode, CNT = count

Explanation: An error was encountered during CLIST processing for reading the parameter member. Most of these errors are caused by having a mismatch between the level of the CLIST and that of the parameter member being read.

The 'errcode' is an error code defined for the CLIST. The error code values and associated meanings are found in *TSO/E CLISTS*.

The 'count' value is the number of parameter values already read.

This message is issued by the following installation CLIST: DSNTINST

System action: The CLIST will terminate.

User response: Be sure the parameter member and the

CLIST are at the same level.

Problem determination: If you suspect an error in the process, set CONTROL(CONLIST) and invoke the CLIST in batch, so that the tracing can be sent with the error documentation. You will also need the listing of the CLIST and the parameter member being read.

DSNT456I ERROR IN VARIABLE varname, VALUE IS value

Explanation: The variable named 'varname' does not have a valid value. Its current value is 'value'. Subsequent messages will describe the reason the value is invalid and the action taken by the CLIST.

If the error is in the CONTROL value, the value was specified when the CLIST was invoked. The value is not valid, and no additional tracing will be done. The permitted values are NONE, LIST, CONLIST, and SYMLIST. Each value produces more CLIST tracing information than those which precede it. If the error is in other values, the value was specified when the CLIST was invoked on the ISPF panels.

This message is issued by the following installation CLIST: DSNTINST

Severity: 8 (error)

System action: The CLIST will continue, but will not produce jobs or parameter data.

User response: Check the subsequent messages. Exit, then invoke the CLIST with a valid value for the variable.

Problem determination: If you suspect an error in the process, set CONTROL(CONLIST) and invoke the CLIST in batch, so that the tracing can be sent with the error documentation. The CLIST itself and the parameter member read in should be sent as well.

DSNT457I The SUBSYSTEM NAME MUST BE DIFFERENT THAN THE PROC NAME

Explanation: Internal MVS restrictions require that the IRLM subsystem name be different than the procedure name.

This message is issued by the following panel: DSNTIPI

System action: The panel is redisplayed and the cursor is placed on the field in error.

User response: Change either the SUBSYSTEM NAME or the PROC NAME so these fields are unique with respect to each other.

DSNT458I VALUE MUST BE NUMERIC

Explanation: The variable named in message DSNT456I must be an integer. Only digits are allowed, with no decimal point or fractions. If the value is a number suffixed by K or M, it is interpreted as the indicated number multiplied by 1024 or 1048676. If any other non-numeric character is present in the value, or if the K or M is not the final character, this message is produced.

This message is issued by the following installation CLIST: DSNTINST

Severity: 8 (error)

System action: The CLIST will continue, but will not produce edited jobs or parameter data.

User response: This CLIST will continue, but will not produce edited jobs. The value of the variable has been set to the value shown in message DSNT465I. You will need to reenter the output parameter and output data set prefix. Examine the value and reset it if needed.

DSNT459I VALUE IS TOO LOW, MINIMUM IS min

Explanation: The variable named in message DSNT456I must be larger than its current value. The minimum value allowed is 'min'.

This message is issued by the following installation CLIST: DSNTINST

Severity: 8 (error)

System action: The CLIST will continue, but will not produce edited jobs. The CLIST resets the incorrect value to 'min'.

User response: See message DSNT465I for the CLIST action.

DSNT460I VALUE IS TOO HIGH, MAXIMUM IS max

Explanation: The variable named in message DSNT456I must be smaller than its current value. The maximum value allowed is 'max'.

This message is issued by the following installation CLIST: DSNTINST

Severity: 8 (error)

System action: The CLIST will continue, but will not produce edited jobs. The CLIST resets the incorrect value to 'max'.

User response: See message DSNT465I for the CLIST action.

DSNT461I THE BUFFER POOL MAXIMUM IS LESS THAN THE MINIMUM VALUE ENTERED

Explanation: Buffer pool maximums must be greater than or equal to the minimum value for the same buffer pool.

This message is issued by the following panel: DSNTIPE

System action: The panel is redisplayed and the cursor is placed on the field in error.

User response: Either reduce the minimum value or increase the maximum value for the indicated buffer pool buffer.

DSNT462I VALUE IS TOO LONG, MAXIMUM IS length CHARACTERS

Explanation: The variable named in message DSNT456I must be shorter than its current value. The maximum length allowed is 'length'.

This message is issued by the following installation CLIST: DSNTINST

Severity: 8 (error)

System action: The CLIST will continue, but will not produce edited jobs. The CLIST resets the value to the default value.

User response: See message DSNT465I for the CLIST action.

DSNT463I DATA SHARING FIELD MUST BE BLANK IN UPDATE MODE.

Explanation: This field is not relevant in Update mode and must be blank This message is issued by panel: DSNTIPA1

System action: The panel is redisplayed and the cursor is placed on t field in error.

User response: Replace the value with a blank.

DSNT464I VALUES PERMITTED ARE values

Explanation: The variable named in message DSNT456I must be one of the 'values' listed.

This message is issued by the following installation CLIST: DSNTINST

Severity: 8 (error)

System action: The CLIST will continue, but will not produce edited jobs.

User response: See message DSNT465I for the CLIST action.

DSNT465I VARIABLE varname SET TO value

Explanation: The variable, 'varname', named in message DSNT456I is in error. Possible reasons for the error are listed in messages: DSNT458I, DSNT459I, DSNT460I, DSNT462I, DSNT463I, DSNT464I.

This message is issued by the following installation CLIST: DSNTINST

System action: The CLIST will continue, but will not produce edited jobs. The value has been set to the value shown.

User response: Specify the output data again on the primary install panel. You will be able to see and change the value. If this value is not acceptable, change it, noting the limits.

DSNT466I Output member CANNOT BE DSNTIDVB OR DSNTIDXA

Explanation: The OUTPUT MEMBER NAME field must not be DSNTIDVB or DSNTIDXA. These two members are shipped with DB2 and must not be overwritten.

This message is issued by the following panel: DSNTIPA1

System action: The panel is redisplayed and the cursor is placed on the field in error.

User response: Change the value of the field under the cursor.**Attention:** If the member you specify already exists, it will be overwritten.

DSNT467I OUTPUT MEMBER SHOULD BEGIN WITH DSNTID

Explanation: The output parameter member will be placed into *prefix*.SDSNSAMP. This data set already has a large number of members. To avoid replacing a member that you need, a naming convention with prefix 'DSNTID' is strongly suggested. The suffixes 'VB' and 'XA' are not allowed (see message DSNT466I).

This message is issued by the following installation CLIST: DSNTINST

Severity: 4 (warning)

System action: The CLIST will continue.

User response: Specify a different name for the output member.

DSNT468I ERROR(S) ENCOUNTERED, OUTPUT IS BYPASSED

Explanation: An error was detected earlier in this run. Because of the error, data set editing was not performed.

This message is issued by the following installation CLIST: DSNTINST

Severity: 8 (error)

System action: The CLIST will continue, but will not produce edited jobs.

User response: Correct the errors noted earlier, and specify the output prefix name again to produce the edited data.

DSNT469I INPUT PARAMETERS MUST BE READ BEFORE MIGRATION

Explanation: The INPUT MEMBER NAME on panel DSNTIPA1 must have a value to allow migration values to be read. This is to permit the new values to be initialized.

This message is issued by the following installation CLIST: DSNTINST

Severity: 8 (error)

System action: The CLIST will return to the main install panel, DSNTIPA1.

User response: Specify DSNTIDXA as the INPUT MEMBER NAME and specify the current value for the migration member.

DSNT470I WARNING, ONLY ONE VOLUME WAS SPECIFIED

Explanation: Only one volume serial number was specified on the install panel or CLIST invocation. This does not allow adequate recovery or performance unless it is intended to be a single volume test system.

This message is issued by the following installation CLIST: DSNTINST

Severity: 4 (warning)

System action: The CLIST will continue.

User response: If this subsystem is intended to be recoverable, more than one volume is needed. Rerun the installation tailoring process, specifying at least one more volume on the DSNTIPA2 panel. If this is just a single volume test system, be sure that you specify a single copy of the log data sets. The boot strap data sets are small and not active, so that two are recommended even on a single volume system.

DSNT471I BEGINNING PARAMETER INPUT MEMBER inmem

Explanation: The input parameter member, 'inmem', is being read. This data includes values for all of the installation parameters.

This message is issued by the following installation CLIST: DSNTINST

Severity: 0 (successful execution)

System action: The CLIST will continue.

DSNT472I WORK FILE DATABASE NAME CANNOT START WITH DSNDB EXCEPT FOR DSNDB07

Explanation: The work file database name cannot start with the characters DSNDB unless the name is DSNDB07.

This message is issued by the following panel: DSNTIPK

System action: The panel is redisplayed and the cursor is placed on the field in error.

User response: Change the name so that it does not start with DSNDB.

DSNT473I BEGINNING CHECK PHASE

Explanation: This message indicates the following:
- The values entered on the panels are being checked.
- Values which use the K or M suffix are being translated to numbers.

This message is issued by the following installation CLIST: DSNTINST

Severity: 0 (successful execution)

System action: The CLIST will continue.

DSNT474I READING count, varname, datatype, data sharing scope, minval, maxval, currentval

Explanation: The values are being read from the parameter member. The count, the variable name, the data type, the data sharing scope, the minimum value, the maximum value, and the current value are shown. This message occurs when the CONTROL keyword requests a trace.

This message is issued by the following installation CLIST: DSNTINST

Severity: 0 (successful execution)

System action: The CLIST will continue.

DSNT475I CHECK count, varname, type, minval, maxval, currentval

Explanation: This message indicates the following:
- The values read in or entered on the panels are being checked.
- Values which use the K or M suffix are being translated to numbers.

The count, the variable name, the data type, the minimum value, the maximum value, and the current value are shown. This message occurs when the CONTROL keyword requests a trace.

This message is issued by the following installation CLIST: DSNTINST

Severity: 0 (successful execution)

System action: The CLIST will continue.

DSNT476I BEGINNING PARAMETER OUTPUT MEMBER outmem

Explanation: The output parameter member named 'outmem' is being written. This data includes values for all the install parameters.

This message is issued by the following installation CLIST: DSNTINST

Severity: 0 (successful execution)

System action: The CLIST will continue.

DSNT477I OUTPUT count, varname, type, minval, maxval, currentval

Explanation: The values entered on the panels are being written to the output parameter member. The count, the variable name, the data type, the minimum value, the maximum value, and the current value are shown. This message occurs when the CONTROL keyword requests a trace.

This message is issued by the following installation CLIST: DSNTINST

Severity: 0 (successful execution)

System action: The CLIST will continue.

DSNT478I BEGINNING EDITED DATA SET OUTPUT

Explanation: The output jobs and data are being edited to tailor them. Edited JCL output was requested and no severe errors were encountered.

This message is issued by the following installation CLIST: DSNTINST

Severity: 0 (successful execution)

System action: The CLIST will continue.

DSNT479I THE IRLM GROUP NAME AND DB2 GROUP NAME MUST BE DIFFERENT

Explanation: You entered an IRLM group name on panel DSNTIPJ that is the same as the DB2 group name entered on panel DSNTIPK. The names must be different.

This message is issued by the following panel: DSNTIPJ

System action: The panel is redisplayed and the cursor is placed on the field in error.

User response: Either change the DB2 group name entered on panel DSNTIPK or the IRLM group name on panel DSNTIPJ.

DSNT480I NAME MUST BE 1-16 CHARS, BEGIN WITH A-Z, AND NOT INCLUDE #, @, OR $.

Explanation: The location name specified on panel DSNTIPR is invalid. The location name must be 1-16 alphanumeric characters, excluding the alphabetic extenders #, @ or $, and must begin with a letter.

This message is issued by panel: DSNTIPR

System action: The panel is redisplayed and the cursor is placed on the field in error.

User response: Change the location name to meet the allowable criteria.

DSNT481I DSNTINST *data set* DATA SET SIZE MAY BE TOO SMALL

Explanation: The DSNTINST CLIST encountered an arithmetic overflow during calculation of this data set's size.

System action: DSNTINST continues processing and uses 2147483647 for the result of the calculation that caused the overflow. Thus, the size of the data set might be too small.

User response: Check the data set size in installation job DSNTIJIN. If it is too small, increase it.

DSNT482I TOO MANY CONCURRENTLY ALLOCATED DATA SETS, *opnds*

Explanation: The number (*opnds*) calculated for the number of concurrent databases, tables per database, and table spaces per database, is too large. The maximum number of data sets that are allowed is 100,000. However, please note that virtual storage below the 16 MB line may be the limiting factor for releases prior to z/OS Version 1 Release 2. See DSNT436I and DSNT437I for more information.

The number of concurrently allocated data sets is calculated using the formula:

```
OPNDS = (NUMCONDB X NUMTABLE X NUMINDEX) +
(NUMCONDB X NUMTABSP)
```

Where Represents

NUMCONDB
The maximum number of concurrently active databases (DATABASES on panel DSNTIPE)

NUMTABLE
The estimated average number of tables per database (TABLES on panel DSNTIPD)

NUMINDEX
The estimated average number of indexes per table (set to 2)

NUMTABSP
The estimated average number of table spaces per database (TABLE SPACES on panel DSNTIPD)

This message is issued by the following installation CLIST: DSNTINST

Severity: 4 (warning)

System action: The CLIST reduces the value of NUMCONDB so that the calculated value for OPNDS is within the allowed range. CLIST processing continues.

User response: Do one of the following:

- Accept the reduced value for NUMCONDB calculated by the CLIST.
- Reenter smaller values for NUMCONDB, NUMTABLE, or NUMTABSP on the SIZES and STORAGE SIZES panels. You can do this by either using the END key to go back to those panels or by running the CLIST again.

DSNT483I NAME MUST BE 1-8 CHARS AND MUST BEGIN WITH A-Z

Explanation: The DB2 data sharing group name specified on panel DSNTIPK or the IRLM data sharing group name specified on panel DSNTIPJ is invalid. The DB2 or IRLM group name must be 1-8 characters, and must begin with a letter.

This message is issued by panels: DSNTIPJ, DSNTIPK

System action: The panel is redisplayed and the cursor is placed on the field in error.

User response: Change the group name to meet the allowable criteria.

DSNT485I type SIZE = size K

Explanation: This is the estimated main storage size for part of the DB2 database address space. This is the most constrained address space. The 'type' can be BUFFER POOL, SORT POOL SIZE, RID POOL SIZE, EDMPOOL STORAGE, DATA SET STORAGE, CODE STORAGE, or WORKING STORAGE.

The storage for buffer pools, sort pools, rid pools, EDMPOOL, and code is above the 16MB line. Most of the main storage for data set control blocks is below the 16MB line and is included in the region size. Working storage includes storage above and below the 16MB line, but storage above the line is not included in the region. Refer to Part 2 of *DB2 Installation Guide* for more information regarding this calculation.

This message is issued by the following installation CLIST: DSNTINST

Severity: 0 (successful execution)

System action: The CLIST continues.

DSNT486I TOTAL MAIN STORAGE = totregion K

Explanation: This is the estimated total region size. It includes storage above the 16MB line. Refer to Part 2 of *DB2 Installation Guide* for more details regarding this calculation.

This message is issued by the following installation CLIST: DSNTINST

Severity: 0 (successful execution)

System action: The CLIST continues.

DSNT487I TOTAL STORAGE BELOW 16MB = region K (ASSUMING SWA IS ABOVE THE LINE)

Explanation: This message provides an estimate of the virtual storage required below the 16MB line to check for virtual storage constraints. This calculation presumes that you have moved the SWA for DB2 above the 16MB line. Refer to Part 2 of *DB2 Installation Guide* for details regarding this calculation.

This message is issued by the following installation CLIST: DSNTINST

Severity: 0 (successful execution)

System action: The CLIST continues.

User response: Check for virtual storage constraints. Your MVS should permit at least this amount of virtual storage below the 16MB line.

DSNT488I VOLUME voln WILL REQUIRE AT LEAST nn 4K BLOCKS

Explanation: This message states the number of 4KB blocks of space that will be required for the specified volume in order to install or migrate the current release of DB2.

Volume 'voln' is specified in the VOLUME SERIAL fields on install panel DSNTIPA2, or with the VOLSDATn input parameters to the Install CLIST. The message indicates how much space will be needed on the selected volume to install or migrate this release of DB2.

This message is issued by the following installation CLIST: DSNTINST

Severity: 0 (successful execution)

System action: Processing continues.

User response: Ensure that the volume has at least as much available storage as the message indicates it should have. If it does not, rerun the INSTALL CLIST specifying different volumes, or create more available space on the originally specified volume.

In order to convert the number of 4KB blocks required to the number of cylinders required, use the table below. Divide the number of 4KB blocks (in the

DSNT488I message) by the number of 4KB blocks per cylinder (in the table) to get the number of cylinders required.

Device Type	4KB Blocks per Cylinder	4KB Blocks per Track
3330	57	3
3340	24	2
3350	120	4
3375	96	8
3380	150	10

These size estimates are for system data only. They do not include: user data, archive logs, image copies, or temporary work data sets other than the temporary database.

DSNT489I CLIST EDITING *dsname(member), explanation*

Explanation: The CLIST is editing the data set and member *dsname(member)*. The *explanation* is the type of data being edited and has one of these values:

INSTALL JCL
Jobs that are edited and run as directed during the installation process.

MIGRATE JCL
Jobs that are edited and run as directed only during the migration process.

FALL BACK JCL
A job that is only run as directed when falling back to the previous release.

RECOVERY JCL
A job that is only run as directed during a recovery process. It deletes the DB2 data sets.

SAMPLE JCL
Jobs that are edited and run during installation verification to add the sample application.

SAMPLE DATA
Edited data that installation verification jobs require.

CLIST Customized CLISTs to reflect your names used during the installation process.

This message is issued by the following installation CLISTs:

DSNTINST DSNTINS1 DSNTINS2

Severity: 0 (successful execution)

System action: The CLIST continues.

DSNT490I SAMPLE DATA UNLOAD PROGRAM

Explanation: This is the header, indicating a normal start for the application program.

DSNT491I TOO MANY TABLES ENTERED (MAXIMUM OF 100 TABLES)

Explanation: The number of tables entered for unloading exceeded the maximum of 100 tables. The first 100 tables in the input should have been processed.

System action: The application program is terminated.

User response: Split the run into multiple runs, none with more than 100 tables.

Problem determination: Check the application program input. Make sure there are no more than 100 tables.

DSNT492I SQL WARNING DURING SQL STATEMENT func, TABLE tttttttt

Explanation: The SQL statement may not be executed correctly. The function 'func' may be PREPARE, OPEN, FETCH, or CLOSE, indicating the type of SQL statement encountering the error. Table 'tttttttt' is the name of the table or view being unloaded.

System action: The request may not be executed correctly.

User response: Refer to *DB2 Codes*.

Problem determination: Determine which SQL statement in the application program caused the warning and see if it is reproducible. Follow the diagnosis approach suggested for the warning.

DSNT493I SQL ERROR DURING SQL STATEMENT func, TABLE tttttttt

Explanation: The SQL statement may not be executed correctly. The function 'func' may be PREPARE, OPEN, FETCH, or CLOSE, indicating the type of SQL statement encountering the error. Table tttttttt is the name of the table or view being unloaded.

System action: The request was not executed correctly.

User response: Refer to *DB2 Codes*.

Problem determination: Determine which SQL statement in the application program caused the error and see if it is reproducible. Follow the diagnosis approach suggested for the SQL return code.

DSNT494I ERROR DURING OPEN OF DDNAME SYSRECnn

Explanation: The ddname listed, where 'nn' ranges from 00 to 99, could not be successfully opened. A

possible reason is that the DD statement to the data set was not present in the JCL.

System action: The application program is terminated.

User response: Check the JCL to ensure that this DD statement is provided and is able to be used for output.

Problem determination: Make sure that the same number of DD statements and table names were provided.

DSNT495I SUCCESSFUL UNLOAD n ROWS OF TABLE tttttttt

Explanation: The message indicates that 'n' rows of table or view 'tttttttt' were unloaded. The 'n' rows were counted during the unload processing.

System action: The request was executed successfully.

DSNT496I UNRECOGNIZED DATA TYPE CODE OF nnnnn

Explanation: The PREPARE returned an invalid data type code of decimal 'nnnnn'.

System action: The next table is processed.

User response: Notify the system programmer.

System programmer response: See the Problem Determination section.

Problem determination: There is an error in the unload program or a new data type has been added. The recognized data types are fullword integer, halfword integer, fixed-length character, variable length character, packed decimal, floating-point, fixed-length graphic, variable length graphic, and long, variable length graphic.

DSNT497I RETURN CODE FROM MESSAGE ROUTINE DSNTIAR

Explanation: The message formatting routine detected an error.

System action: The application program is terminated.

User response: Notify the system programmer.

System programmer response: See the Problem Determination section.

Problem determination: There is an error in the unload program. The return code from DSNTIAR is given in the MVS output of your job. When reporting this problem, have the return code, the application program listing, and JCL available.

DSNT498I ERROR, NO VALID COLUMNS FOUND

Explanation: The PREPARE returned data which did not produce a valid output record. The data was not recognized as valid, so the columns were not unloaded.

System action: The next table is processed.

User response: Notify the system programmer.

System programmer response: See the Problem Determination section.

Problem determination: There is an error in the unload program. When reporting this problem, have the application program listing and its JCL available.

DSNT499I ERROR IN INPUT PARAMETER LIST

Explanation: The parameter list specified when DSNTIAUL or DSNTEP2 was invoked contains one or more invalid values.

System action: The application program is terminated.

User response: Notify the system programmer.

System programmer response: Check the parameter values entered when DSNTIAUL or DSNTEP2 was invoked. For DSNTIAUL, the correct parameter value is 'SQL'. This parameter indicates that DSNTIAUL input is in complete SQL statements terminated by a semicolon. For DSNTEP2, the correct parameter values are either 'ALIGN(LHS)', 'ALIGN(MID)', 'MIXED', or 'NOMIXED'. The first two parameter values represent the two output alignment options available in DSNTEP2 where 'ALIGN(LHS)' will align the output on the left margin and 'ALIGN(MID)' will center the output. The remaining two parameter values designate whether or not DSNTEP2 will recognize mixed character data where 'MIXED' means that it will and 'NOMIXED' means that it will not recognize mixed character data.

DSNT500I *csect-name* **RESOURCE UNAVAILABLE REASON** *reason* **TYPE** *type* **NAME** *name*

Explanation: The requested function requires a resource that is currently unavailable. In conjunction with other messages this message will identify the system action and the action that should be taken by the installation or operator.

'type' and 'name' identify the resource that is currently unavailable. The type code identifies the type, format, and content of the resource named.

Operator response: Note the reason code, the type code, and the name. See the Problem Determination section for an explanation of the collected data.

Problem determination: The reason the resource is unavailable is identified by a reason code. The reason codes that appear in this message are described in *DB2*

Codes. Use Table 2 in Appendix A, "Problem determination," on page 659 to find the type of resource identified in the message.

If type is X'D00' or X'D01', the table name can be determined by using the following SQL statement:

```
SELECT CREATOR, NAME
FROM SYSIBM.SYSTABLES
WHERE DBID = dbid AND OBID = obid;
```

However, if the failing table has been dropped, then the above SELECT fails or possibly might return an incorrect name.

Collect the following diagnostic item listed in Appendix A, "Problem determination," on page 659: 1.

DSNT501I *csect-name* **RESOURCE UNAVAILABLE**

CORRELATION-ID	= id1
CONNECTION-ID	= id2
LUW-ID	= id3
REASON	= reason
TYPE	= type
NAME	= name

Explanation: The requested function requires a resource that is currently unavailable. This message is issued with other messages that identify the system action and the action the installer or operator should take.

DB2 can recover from a resource not available in some cases and can complete successfully. One example is during setup of a performance enhancement during bind time for execution time. If DB2 recovers successfully, then no action is necessary.

For type=905, reason=00C90092, and name=IRLM, message DSNT501I is usually issued once. It is suppressed at the DB2 subsystem level until the IRLM out-of-storage condition is relieved.

id1 The correlation identifier of the thread associated with the requester of the resource.

id2 The connection identifier of the requester.

id3 The logical-unit-of-work identifier (LUW-ID) of the thread associated with the requester of the resource. An asterisk (*) indicates that the thread does not have a LUW-ID. If the LUW-ID is not an asterisk, the value is a logical-unit-of-work-id assigned to the thread. This is followed by an equal sign (=) and a token value, which can be used in place of logical-unit-of-work-id in any DB2 command that accepts logical-unit-of-work-id as input.

type The type of resource that is currently unavailable. The 'type' identifies the type, as well as the format and content, of the resource 'name'. Use Table 2 in Appendix A, "Problem determination," on page 659 to find the type of resource identified in the message.

name The name of the resource that is currently unavailable.

reason The reason code that tells why the resource is unavailable.

Operator response: Note the 'reason' code, the 'type' code, and the 'name'.

System programmer response: To perform the
operation on the object, you must migrate the catalog
to the current release. To do this, run DSNTIJTC. Refer
to Part 2 of *DB2 Installation Guide* for information about
this job.

Problem determination: A type X'401' indicates that a cascading revoke of SYSADM or SYSCTRL in a lower-level release attempted to revoke privileges or grants on objects that are available only in a higher-level release. Because the cascading revoke affects objects that are unknown to the lower-level release, you can only revoke SYSADM or SYSCTRL from the higher-lever release.

Collect the following diagnostic item listed in Appendix A, "Problem determination," on page 659: 1.

DSNT502I ERROR IN DSNTIAUL SQL INPUT: statement

Explanation: The DSNTIAUL input SQL statement could not be prepared.

System action: The application program terminates.

User response: Notify the system programmer.

System programmer response: Check that the DSNTIAUL input statement is a complete SQL statement, ending with a semicolon.

DSNT503I UNLOAD DATA SET *ddname* **RECORD LENGTH SET TO** *nnnnn***.**

Explanation: If the ddname is SYSRECnn, the record length specified in the JCL for the data set is not large enough to contain an entire row of unloaded data, or the record length was not specified in the JCL.

If the ddname is SYSPUNCH, the record length specified in the JCL is not between 80 and 255, or the record length was not specified in the JCL.

System action: DB2 sets the record length of the unload data set to the value specified in the message text.

DSNT504I UNLOAD DATA SET *ddname* **BLOCKSIZE SET TO** *nnnnn***.**

Explanation: The block size for the SYSRECnn or SYSPUNCH data set was one of the following:

- Not specified in the JCL

- Set to zero in the JCL
- Less than the required record length
- Not an even multiple of the required record length

System action: DB2 sets the block size of the unload data set to the value specified in the message text.

DSNT505I DSNTIAUL OPTIONS USED: 'nnn'.

Explanation: You specified these options when you invoked DSNTIAUL. 'SQL' is the only valid option.

Severity: 0 (informational)

DSNT506I INPUT STATEMENT WAS NOT A FULL SELECT ON A SINGLE TABLE. LOAD STATEMENT WILL NEED MODIFICATION.

Explanation: The SELECT statement used to request a table unload contained one of the following:

- A SELECT clause other than "SELECT * FROM"
- A FROM clause that does not contain exactly one table name

The table to be loaded is therefore not the same as the table from which the data is unloaded. DSNTIAUL is unable to determine the table name and all column names for the table to be loaded.

Severity: 4 (warning)

System action: DSNTIAUL generates a LOAD statement containing the string 'TBLNAME' for the table name. Some column names may be blank.

User response: Change the LOAD statement table name to the name of a table with the indicated column descriptions.

DSNT507I *csect-name* **RECORD LENGTH NEEDED IS GREATER THAN MAXIMUM ALLOWED FOR TABLE** *table-name*

Explanation: In order to unload table *table-name*, DB2 needs an output data set with a record length greater than the system maximum of 32760 bytes. This occurs when the table to be unloaded is in a 32KB page table space and contains a combination of long columns and a number of date, time, or timestamp columns.

Severity: 8 (error)

System action: The application program is terminated.

DSNT508I WARNING - NONSTANDARD JOB CARD (*cardstat*), NO JOB CARDS WILL BE ADDED TO THE EDITED JOBS

Explanation: The job card you entered in field 3 of panel DSNTIPY contains one or more of the following errors:

- Less than 7 characters ('cardstat' = TOO SHORT)
- "//" missing as the first 2 characters ('cardstat' = MISSING //)
- Missing the word "JOB" ('cardstat' = MISSING "JOB"TOKEN)

This message is issued by the following installation CLIST: DSNTINST

System action: The message is written to the screen, after all panels are displayed and before any job editing. No JCL is added to the edited jobs.

User response: You can continue with job editing and manually add JCL to the edited jobs. Alternatively, you can repeat the installation process and make corrections to the job card entered on panel DSNTIPY.

DSNT509I WARNING - MIGRATION INPUT MEMBER LEVEL IS *rel_ind***. LEVEL** *prev_rel* **IS REQUIRED. RETURN TO PANEL DSNTIPA1 TO CHANGE MIGRATION INPUT MEMBER**

Explanation: The migration input member you entered on panel DSNTIPA1 contains incorrect release indicator 'rel_ind'. It should contain the indicator for the prior release ('prev_rel'). If your input member does not contain a release indicator, 'rel_ind' will be <null>. This probably indicates that you are using a down-level member.

This message is issued by the following installation CLIST: DSNTINST

System action: The message is written to the screen, after display of panel DSNTIPA1.

User response: Press ENTER to proceed to the next panel, and then PF3 to return to panel DSNTIPA1 in order to enter an input member with the correct release indicator.

DSNT510I WARNING - INPUT MEMBER *inmem* **LEVEL IS** *rel_ind***. LEVEL** *curr_rel* **IS REQUIRED. RETURN TO PANEL DSNTIPA1 TO CHANGE INPUT MEMBER**

Explanation: The input member 'inmem' you entered in field 6 on panel DSNTIPA1 contains incorrect release indicator 'rel_ind'. It should contain the indicator for the current release ('curr_rel'). If your input member does not contain a release indicator, 'rel_ind' will be <null>. This probably indicates that you are using a down-level member.

This message is issued by the following installation CLIST: DSNTINST

System action: The message is written to the screen, after display of panel DSNTIPA1.

User response: Press ENTER to proceed to the next

panel, and then PF3 to return to panel DSNTIPA1 in order to enter an input member with the correct release indicator.

DSNT511I FIELDS 2 AND 5 MUST BE UNIQUE FOR EACH NEW MEMBER INSTALLED

Explanation: This warning message is issued to remind you that the IRLM SUBSYSTEM and IRLM PROC names must be unique for each new member installed within a DB2 data sharing group. The installation process does not check for uniqueness of these names within a group.

This message is issued by the following panel: DSNTIPI

User response: Enter unique names in these fields for each member of a DB2 data sharing group.

DSNT512I DSNT512I WARNING: ENTER UNIQUE NAMES FOR LUNAME AND LOCATION NAME

Explanation: This warning message is issued to remind you that you must enter values for DB2 LOCATION NAME and DB2 NETWORK LUNAME fields.

LUNAME is used to uniquely identify this DB2 subsystem within the network and to uniquely identify trace records for correlation of accounting records. The NETWORK LUNAME must be unique within the network. The installation process does not check for uniqueness of this name within a data sharing group.

The LOCATION NAME must be unique within the network, but must be the same on all members within a DB2 data sharing group.

This message is issued by the following panel: DSNTIPR

User response: Enter unique names in these fields.

DSNT513I FIELD 2 MUST BE UNIQUE FOR EACH NEW MEMBER INSTALLED

Explanation: This warning message is issued to remind you that the DB2 PROC NAME field on panel DSNTIPX or the RESYNC PORT fieldon panel DSNTIP5 must be unique for each new member installed within a DB2 data sharing group. The installation process does not check for uniqueness of these values within a group.

This message is issued by the following panels: DSNTIPX and DSNTIP5

User response: Enter unique values in these fields for each member of a DB2 data sharing group.

DSNT514I FIELDS 2 AND 3 MUST BE UNIQUE FOR EACH NEW MEMBER INSTALLED

Explanation: This warning message is issued to remind you that the MEMBER NAME and WORK FILE DB NAME must be unique for each new member installed within a DB2 data sharing group. The installation process does not check for uniqueness of these names within a group.

This message is issued by the following panel: DSNTIPK

User response: Enter unique names in these fields for each member of a DB2 data sharing group.

DSNT515I FIELD 7 MUST BE UNIQUE FOR EACH NEW MEMBER INSTALLED

Explanation: This warning message is issued to remind you that the MEMBER IDENTIFIER field on panel DSNTIPJ or the DSNZPARM NAME field on panel DSNTIPO must be unique for each new member installed within a DB2 data sharing group. The installation process does not check for uniqueness of these values within a group.

This message is issued by the following panels: DSNTIPJ and DSNTIPO

User response: Enter unique values in these fields for each member of a DB2 data sharing group.

DSNT516I FIELDS 1, 2, AND 4 MUST BE UNIQUE FOR EACH NEW MEMBER INSTALLED

Explanation: This warning message is issued to remind you that the SUBSYSTEM NAME, COMMAND PREFIX, and SUBSYSTEM SEQUENCE fields must be unique for each new member installed within a DB2 data sharing group. The installation process does not check for uniqueness of these values within a group.

This message is issued by the following panel: DSNTIPM

User response: Enter unique values in these fields for each member of a DB2 data sharing group.

DSNT517I CHARACTERS 2-18 MUST BE A-Z, 0-9, #, @, $, OR UNDERSCORE

Explanation: WLM environment names that do not meet the following criteria are not allowed. A WLM environment name may contain up to eighteen characters. The first character of a WLM environment name must either be an alphabetic character (A-Z) or a standard alphabetic extender (in the United States, the standard alphabetic extenders are '#', '@', '$'). For characters two through eighteen, a character may be an alphabetic character (A-Z), an alphabetic extender ('#',

'@', '$'), a numeric character (0-9), or the underscore character ('_').

This message is issued by panel: DSNTIPX

System action: The panel is redisplayed, and the cursor is placed on the field with the error.

User response: Change the WLM environment name to meet the criteria described above.

DSNT518I THE VALUES FOR DRDA PORT AND RESYNC PORT CANNOT BE THE SAME

Explanation: You entered the same value for DRDA PORT and RESYNC PORT. The values for these fields must be different.

This message is issued by the following panel: DSNTIP5

System action: The panel is redisplayed and the cursor is placed on the field in error.

User response: Change either the DRDA PORT or the RESYNC PORT value so that they are different.

DSNT519I IF YOU SPECIFY EITHER DRDA OR RESYNC PORT, BOTH MUST BE SPECIFIED

Explanation: You entered a value for either DRDA PORT or RESYNC PORT, but not the other. If either DRDA PORT or RESYNC PORT is specified, both must be specified.

This message is issued by the following panel: DSNTIP5

System action: The panel is redisplayed and the cursor is placed on the field in error.

User response: Either enter a value for the null field, or blank out both DRDA PORT and RESYNC PORT.

DSNT520I DB2's default sizes for index and table spaces are increased in V8

Explanation: This message appears on DB2 installation panel DSNTIP7 when the CLIST is in MIGRATE mode and an entry of 0 appears in either field 4 TABLE SPACE ALLOCATION or field 5 INDEX SPACE ALLOCATION (or both). An entry of 0 indicates that DB2 default allocation sizes are to be assigned for table spaces or index spaces created without the USING clause. The purpose of the message is to alert you that the DB2 default allocation sizes have changed in Version 8.

- If you specify 0 for the TABLE SPACE ALLOCATION field, then 1 cylinder will be the default PRIQTY space allocation for non-LOB tablespaces and 10 cylinders will be the default PRIQTY space allocation for LOB tablespaces.
- If you specify 0 for the INDEX SPACE ALLOCATION field, then 1 cylinder will be the default PRIQTY space allocation for indexes.

The DB2 default SECQTY quantity depends on the setting you specify for DSNTIP7 field 7 OPTIMIZE EXTENT SIZING. See the discussion of OPTIMIZE EXTENT SIZING for more information.

Severity: 4

System action: Processing continues.

User response: To assign the new DB2 default allocation sizes to table spaces created without the USING clause, specify 0 in field 4 TABLE SPACE ALLOCATION. Otherwise, specify an explicit size.

To assign the new DB2 default allocation size to index spaces created without the USING clause, specify 0 in field 5 INDEX SPACE ALLOCATION. Otherwise, specify an explicit size.

DSNT526I WARNING: DSNHDECP SPECIFIES AN UNDEFINED (0) VALUE FOR *ccsid-parm* NOTIFY THE SYSTEM PROGRAMMER.

Explanation: The DSNHDECP module that is used by the DB2 precompiler contains an undefined (0) value for the CCSID parameter that is indicated by *ccsid-parm*. The use of CCSIDs that are undefined will be disallowed by DB2 in the near future.

System action: Processing continues.

User response: Notify the System Programmer.

System programmer response: DSNHDECP needs to specify a valid, non-zero default CCSID (Coded Character Set Identifier) for each of the three encoding schemes that are supported: ASCII, EBCDIC, and UNICODE. Processing character data with an undefined CCSID can lead to unpredictable results, including loss or corruption of the data. A default CCSID must be defined even for encoding schemes not used on your DB2 system.

If *ccsid-parm* is ASCCSID, then the default CCSID for single-byte ASCII-encoded character data is undefined (set to 0). If *ccsid-parm* is SCCSID, then the default CCSID for single-byte EBCDIC-encoded character data is undefined (set to 0).

Using the undefined CCSID for EBCDIC will be disallowed in the near future by APAR PQ71079. Using the undefined CCSID for ASCII will be disallowed in a future release of DB2.

You need to take steps now to select an appropriate CCSID for each undefined CCSID, and to plan for and schedule deployment of the updated DSNHDECPs on your DB2 subsystem or data sharing system. See the documentation provided with the PTF for APAR PQ56697 for further guidance.

Do not attempt to choose or change a CCSID without guidance from IBM. Using the wrong CCSID can corrupt your data and render it unrecoverable.

DSNT527I ERROR: DB2 ATTEMPTED TO START USING THE DB2-SUPPLIED DSNHDECP MODULE. NOTIFY THE SYSTEM PROGRAMMER.

Explanation: The DSNHDECP module that is supplied by DB2 is being used to run the DB2 precompiler. DB2 requires site-dependent settings that are specified in the DSNHDECP module that was created when DB2 was installed. The DSNHDECP that was generated during installation module typically is located in the SDSNEXIT library.

System action: DB2 startup is discontinued.

User response: Notify the System Programmer.

System programmer response: Check the STEPLIB, JOBLIB, or system linbklist concatenation used in the DB2 address spaces. To ensure that the proper DSNHDECP module is used, verify that the library that contains the DSNHDECP module that is generated during installation precedes the SDSNLOAD library that contains the DSNHDECP module that was supplied by DB2.

DSNT528I USE PROTECTION MUST BE YES WHEN CACHE DYNAMIC SQL IS YES

Explanation: You entered NO for USE PROTECTION on panel DSNTIPP and YES for CACHE DYNAMIC SQL on panel DSNTIP4. You must enter YES for USE PROTECTION if CACHE DYNAMIC SQL is YES.

System action: The panel is redisplayed, and the cursor is placed on the field that is in error.

User response: If you see this message on panel DSNTIPP, enter a value of YES for USE PROTECTION, or return to panel DSNTIP4 and enter NO for CACHE DYNAMIC SQL. If you see this message on panel DSNTIP4, enter a value of NO for CACHE DYNAMIC SQL, or go to panel DSNTIPP and enter YES for USE PROTECTION.

DSNT530I *program-name* **COULD NOT RUN BECAUSE IT WAS NOT INVOKED FROM AN APF-AUTHORIZED LIBRARY**

Explanation: *program-name* could not run for one of the following reasons:

- *program-name* was not invoked from an APF-authorized library.
- One or more libraries in the STEPLIB or JOBLIB concatenation the job that invokes *program-name* is not APF-authorized.

System action: The program terminates.

User response: Notify the system programmer.

System programmer response: Ensure that *program-name* is link-edited with option AC(1) and resides in an APF-authorized library. Ensure that all libraries in the STEPLIB and JOBLIB concatenation for the job that invokes *program-name* are APF authorized. If *program-name* is an external module for a DB2 stored procedure or user-defined function, ensure that the corresponding SYSIBM.SYSROUTINES entry specifies a WLM-established address space.

DSNT531I *schema.routine-name* **RECEIVED SQLCODE** *sqlcode* **FROM SQL STATEMENT** *sql-statement*

Explanation: When *schema.routine-name* executed *sql-statement*, DB2 returned the SQL warning or error *sql-code*.

System action: *schema.routine-name* terminates.

User response: Notify the system programmer.

System programmer response: Check the messages in the SYSPRINT data set for the stored procedures address space where routine *schema.routine-name* runs. Message DSNT408I provides specific information about the SQL warning or error. Refer to *DB2 Codes* for more information on the SQL error.

DSNT532I *schema.routine-name* **RECEIVED SAF RETURN CODE** *return-code* **FROM RACROUTE COMMAND** *command***.**

Explanation: When *schema.routine-name* executed RACROUTE command *command*, the OS/390 System Authorization Facility (SAF) returned an unexpected code *return-code*.

System action: *schema.routine-name* terminates.

User response: Notify the system programmer.

System programmer response: Refer to the *z/OS Security Server RACROUTE Macro Reference* for more information about the SAF error.

DSNT533I *schema.routine-name* **RECEIVED RETURN CODE** *return-code* **FROM MVS SERVICE** *service-name***, FUNCTION** *function-name***.**

Explanation: *schema.routine-name* executed MVS service *service-name* for function *function-name* and received an unexpected code *return-code*.

System action: *schema.routine-name* terminates.

System programmer response: Refer to the discussion of service *service-name* in the *z/OS MVS Programming: Authorized Assembler Services Reference Volumes 1-4* for assistance with diagnosing the error.

User response: Notify the system programmer.

DSNT534I *schema.routine-name* **RECEIVED AN ALERT FROM THE EXTENDED MCS CONSOLE:** *alert-type***.**

Explanation: *schema.routine-name* has activated an extended MCS console to monitor system messages. This console has posted an alert as indicated by *alert-type.*

System action: *schema.routine-name* terminates.

User response: Notify the system programmer.

System programmer response: If *routine-name* is the DB2-supplied stored procedure WLM_REFRESH then this message indicates an error in program DSNTWR. Refer to the *z/OS MVS Programming: Authorized Assembler Services Guide*, the subtopic entitled "What to Do if Message Queuing Stops in the discussion of What Is an Extended MCS Console?" for help. If this information does not help you, contact IBM for assistance.

DSNT540I *wlm-environment-name* **WAS REFRESHED BY** *user* **USING AUTHORITY FROM SQL ID** *sqlid*

Explanation: DB2 stored procedure WLM_REFRESH refreshed WLM environment *wlm-environment-name*. User ID *user* issued the request, using authority from SQL ID *sqlid*.

System action: WLM environment *wlm-environment-name* is refreshed.

User response: None

System programmer response: None

DSNT541I **WLM REFRESH REQUEST FAILED. RESOURCE PROFILE** *subsystem-name***.WLM_REFRESH.***wlm-environment-name* **IS UNDEFINED**

Explanation: DB2 stored procedure WLM_REFRESH rejected a request to refresh WLM environment *wlm-environment-name* because SAF resource profile *subsystem-name*.WLM_REFRESH.*wlm-environment-name* does not exist. WLM_REFRESH uses the resource profile to verify whether the requestor is authorized to refresh the WLM environment. The requestor was connected to DB2 subsystem *subsystem-name*.

System action: WLM_REFRESH rejects the refresh request.

User response: Notify the system programmer.

System programmer response: Create the SAF resource profile in resource class DSNR, and permit READ access to all primary and secondary authorization IDs that need to refresh the WLM environment.

DSNT542I **SQL ID** *sqlid* **IS NOT AUTHORIZED TO REFRESH WLM ENVIRONMENT** *wlm-environment-name*

Explanation: DB2 stored procedure WLM_REFRESH rejected a request to refresh WLM environment *wlm-environment-name* because *sqlid* is not authorized. *sqlid* is the current SQLID of the requestor.

System action: WLM_REFRESH rejects the refresh request.

User response: Notify the system programmer.

System programmer response: Give *sqlid* READ access or higher on the appropriate SAF resource profile. That profile resides in SAF resource class DSNR and has the format *ssid*.WLM_REFRESH.*wlm-environment-name*.

DSNT543I **WLM REFRESH REQUEST FAILED: PARAMETER** *parameter* **IS NULL, BLANK, OR CONTAINS INVALID CHARACTERS**

Explanation: The argument passed to WLM_REFRESH for *parameter* cannot be used for one of the following reasons:

- It is null or blank
- It contains characters that are not allowed for a SAF resource profile or for a DB2 subsystem name

System action: The program terminates.

User response: Notify the system programmer.

System programmer response:

- For the WLM_ENV_NAME parameter, specify a non-null, non-blank argument. That argument must conform to the rules for a SAF resource profile. See Appendix A of *z/OS Security Server RACF Auditor's Guide* for more information.
- For the SSID parameter, specify a null or blank argument, or a value that conforms to the rules for a DB2 subsystem name, as documented in the *DB2 Installation Guide*. If you specify a null or blank value, DB2 determines the subsystem name.

DSNT544I **SQLID** *sqlid* **IS UNDEFINED OR** *user* **IS NOT PERMITTED TO USE IT TO REFRESH** *wlm-environment-name*

Explanation: DB2 stored procedure WLM_REFRESH rejected a request by user id *user* to refresh WLM environment *wlm-environment-name* using authorization id *sqlid*. The request was denied either because *sqlid* does not exist or because *user* is not permitted to use it.

System action: WLM_REFRESH rejects the refresh request.

User response: Notify the system programmer.

System programmer response: If *sqlid* exists, permit

user to use it. Otherwise, create *sqlid*, give it READ access or higher on the appropriate SAF resource profile (which resides in SAF resource class DSNR and has the format *ssid*.WLM_REFRESH.*wlm-environment-name*), and permit *user* to use it.

DSNT545I WLM REFRESH REQUEST FAILED BECAUSE OF AN UNEXPECTED SYSTEM MESSAGE WITH IDENTIFIER *message-identifier*

Explanation: The system has denied a WLM refresh request passed by DB2 stored procedure WLM_REFRESH. *message-identifier* is the identifier of a system message that was issued when the request failed.

System action: The refresh request is denied.

User response: Notify the system programmer.

System programmer response: Examine the system log in the vicinity of the failed refresh request and locate message identifier *message-identifier*. Use this message and related messages to diagnose and correct the problem that is preventing the WLM refresh request.

DSNT546I WLM REFRESH REQUEST CANCELLED BECAUSE AN EXTENDED MCS CONSOLE COULD NOT BE ACTIVATED WITHIN *n* **SECONDS**

Explanation: DB2 stored procedure WLM_REFRESH rejected a request because it was unable to activate an extended MCS console within the allowed period of *n* seconds. The extended MCS console serializes the WLM_REFRESH requests and verifies that z/OS completed the refresh request.

System action: WLM_REFRESH rejects the refresh request.

User response: If you receive DSNT456I from subsequent calls to WLM_REFRESH, notify the system programmer.

System programmer response: This message indicates that WLM_REFRESH is busy processing one or more previous requests. Issue the DB2 command `-DIS PROC(SYSPROC.WLM_REFRESH)` and examine the ACTIVE, QUEUED, and MAXQUE values to analyze whether or not a large backlog of calls to WLM_REFRESH currently exists. To give WLM_REFRESH more time to respond to these requests, you can increase the wait time by updating the setting of the WAITLMT constant in the source code of the external module for WLM_REFRESH, DSNTWR. You need to prepare and bind DSNTWR again for the change to take effect.

DSNT547I THE RESULT OF THE WLM REFRESH REQUEST IS UNKNOWN BECAUSE THE SYSTEM DID NOT REPLY WITHIN *n* **SECONDS**

Explanation: The DB2-supplied stored procedure, WLM_REFRESH, accepted and validated the request to refresh a WLM environment and passed it to z/OS for processing. However, z/OS did not respond within the wait period of *n* seconds.

System action: WLM_REFRESH terminates with return code 4.

User response: Notify the system programmer.

System programmer response: This message indicates that your system is suppressing one or more of the messages that can be generated in response to the VARY WLM command. DSNTWR, the external module for WLM_REFRESH, uses an extended MCS console (also named DSNTWR) to monitor whether or not the VARY WLM command succeeded or failed. Messages with the IDs IWM032I (refresh complete) and IWM031I (refresh in progress) indicate that the refresh request processed successfully. So, for WLM_REFRESH to function correctly, you need to ensure that the system does not suppress messages IWM031I and IWM032I.

This error can also occur if the specified WLM environment requires more than *n* seconds to refresh. In this case, you need to increase the wait time allowed for the reply by updating the setting of the WAITLMT constant in the source code of the external module for WLM_REFRESH, DSNTWR. You need to prepare and bind DSNTWR again for the change to take effect.

DSNT550I *'data-set-name'* **IS UNAVAILABLE FOR CLIST OUTPUT BECAUSE** *reason*

Explanation: Data set *data-set-name* is required for output by the DB2 Install CLIST, but cannot be allocated for the reason that is indicated by *reason*. Possible reasons are:

- IT CANNOT BE ALLOCATED OR OPENED FOR OUTPUT

  An error occurred during an allocate, open, or close operation on *data-set-name.*
- ERROR PROCESSING REQUESTED DATA SET

  The request for *data-set-name* could not be processed.
- MEMBER SPECIFIED, BUT DATASET IS NOT PARTITIONED

  data-set-name already exists, but is not a partitioned data set as required by the CLIST.
- PROTECTED DATASET

  Your userid is not authorized to use *data-set-name*.
- VOLUME NOT ON SYSTEM

  data-set-name is registered in the system catalog on a DASD or tape volume which is currently unmounted or unavailable.

- UNAVAILABLE DATASET
  data-set-name is currently allocated by another user.

System action: The Install panel for specifying *data-set-name* is redisplayed. You cannot proceed to the next DB2 Install panel until the problem is resolved.

User response: Take one of the following actions according to *reason*:

- For errors allocating, opening, or processing *data-set-name*, specific diagnostic messages are displayed by TSO. If you are not seeing these messages, enter the command TSO PROFILE WTPMSG in the command line of the current DB2 install panel. Consult the appropriate MVS system messages and codes manual for further information about these messages.
- If *data-set-name* is indicated as a non-partitioned data set, verify that you have entered the intended data set name for CLIST output. If so, rename or delete the existing data set.
- Consult your system security administrator to obtain access to a protected data set.
- In the event of an unavailable data set, ask the current user(s) to release it. Ask your system programmer to identify the user(s) if you are unable to do so. Otherwise, try later when the data set has been freed.

DSNT551I *csect-name bind-type* **FOR** *object-type* **=** *object-name* **FAILED BECAUSE CHARACTER CONVERSION FROM CCSID** *from-ccsid* **TO CCSID** *to-ccsid* **FAILED WITH ERROR CODE** *error-code* **FOR TABLE** *dbid.obid* **COLUMN** *column-number***.**

Explanation: A conversion error occurred during the conversion of a string to a different coded character set.

bind-type
identifies the type of bind subcommand: BIND, REBIND or FREE.

object-type
identifies the type of the object being bound: PLAN or PACKAGE.

object-name
if 'object-type' is PLAN, 'object-name' is the name of the plan; otherwise, it is the name of the package in 'location.collection.package' format

from-ccsid
identifies the coded character set of the string to be converted.

to-ccsid
identifies the coded character set to which it must be converted.

error-code
indicates the type of error as defined as follows:

8 Length exception (for example, expansion required for PC MIXED data exceeds the maximum length of the string).

12 Invalid code point.

16 Form exception (for example, invalid MIXED data).

20 Translate procedure error (for example, an error during Unicode/390 stage 2 conversions or an exit set the length control field of the string to an invalid value).

24 The conversion services have not been setup correctly or the conversion is not supported.

dbid.obid
gives the database id and the table obid of the object with the conversion error. This could be either the target or the source of the conversion.

column-number
gives the column number of the failing object. This could be either the target or the source of the conversion.

Severity: 8

System action: The *bind-type* operation is not performed.

User response: If the *error-code* is 24 and the conversion request is correct, refer to the section entitled "Character conversion" of *DB2 Installation Guide* for information on how to add conversion support. For all other *error-codes*, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNT552I *csect-name* **CHARACTER CONVERSION FROM CCSID** *from-ccsid* **TO CCSID** *to-ccsid* **FAILED WITH ERROR CODE** *error-code*

Explanation: A conversion error occurred during the conversion of a string to a different coded character set.

from-ccsid
identifies the coded character set of the string to be converted.

to-ccsid
identifies the coded character set to which it must be converted.

error-code
indicates the type of error as defined as follows:

8 Length exception (for example, expansion required for PC MIXED data exceeds the maximum length of the string).

12 Invalid code point.

16 Form exception (for example, invalid MIXED data).

20 Translate procedure error (for example, an error during Unicode/390 stage 2 conversions or an exit set the length control field of the string to an invalid value).

24 The conversion services have not been setup correctly or the conversion is not supported.

Severity: 8

System action: The operation is not performed.

User response: If the *error-code* is 24 and the conversion request is correct, refer to the section entitled "Character conversion" of *DB2 Installation Guide* for information on how to add conversion support. The MVS Command D UNI,ALL can be used to see the conversions that have been defined. For all other *error-codes*, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNT570I WARNING: THIS APPLICATION HAS SELECTED DSNTIAR TO PERFORM SQL MESSAGE FORMATTING BECAUSE THE GET DIAGNOSTICS SERVICE RETURNED AN UNEXPECTED SQLCODE WHILE FORMATTING MESSAGES FOR ONE OR MORE SQL WARNING OR ERROR CONDITIONS. A DUMP OF THE SQLCA FOR THE GET DIAGNOSTICS REQUEST FOLLOWS:

```
SQLCODE: sqlcode          SQLSTATE: sqlstate    SQLERRP: sqlerrp
SQLERRMC: sqlerrmc
SQLERRD1: sqlerrd1        SQLERRD2: sqlerrd2    SQLERRD3: sqlerrd3
SQLERRD4: sqlerrd4        SQLERRD5: sqlerrd5    SQLERRD6: sqlerrd6
SQLWARN0: w0 1: w1 2: w2 3: w3 4: w4 5: w5 6: w6 7: w7 8: w8 9:
          w9 A:wA
```

Explanation: The application program called the GET DIAGNOSTICS service to format one or more of the warning or error conditions that occurred during the processing of the current SQL statement that was input into the system. However, the GET DIAGNOSTICS service also returned warnings or errors, as indicated by the SQLCA dump. The application therefore will use the DSNTIAR message formatting service instead of GET DIAGNOSTICS. **Attention:** Messages formatted by DSNTIAR might contain truncated tokens. Also, when multiple unexpected SQLCODEs were returned for the current inputted SQL statement, DSNTIAR reports on the first one.

System action: The application program continues.

User response: Notify the system programmer.

System programmer response: See the 344 section.

Problem determination: Use the SQLCA dump provided by DSNT570I to determine the cause of the problem. After the problem is resolved, run the application again.

DSNT571I ERROR: THIS APPLICATION IS TERMINATING BECAUSE THE GET DIAGNOSTICS SERVICE RETURNED AN UNEXPECTED SQLCODE WHILE FORMATTING MESSAGES FOR ONE OR MORE SQL WARNING OR ERROR CONDITIONS FROM A ROWSET FETCH STATEMENT OR A NON-ATOMIC STATEMENT. A DUMP OF THE SQLCA FOR THE GET DIAGNOSTICS REQUEST FOLLOWS:

```
SQLCODE: sqlcode          SQLSTATE: sqlstate    SQLERRP: sqlerrp
SQLERRMC: sqlerrmc
SQLERRD1: sqlerrd1        SQLERRD2: sqlerrd2    SQLERRD3: sqlerrd3
SQLERRD4: sqlerrd4        SQLERRD5: sqlerrd5    SQLERRD6: sqlerrd6
SQLWARN0: w0 1: w1 2: w2 3: w3 4: w4 5: w5 6: w6 7: w7 8: w8 9:
          w9 A:wA
```

Explanation: The application program called the GET DIAGNOSTICS service to format one or more of the warning or error conditions that occurred during the processing of a ROWSET FETCH statement or a non-atomic statement. However, the GET DIAGNOSTICS request failed. Since the severity of the original warning or error conditions cannot be determined, the application terminates after reporting this error.

System action: The application program is terminated.

User response: Notify the system programmer.

Problem determination: Use the SQLCA dump provided by DSNT571I to determine the cause of the problem. After the problem is resolved, run the application again.

DSNT601I THE DESCRIPTOR FOR type name DOES NOT EXIST IN DBD01

Explanation: This message is issued for multiple-line WTO. The database, table space, or index space 'name' is in the DB2 catalog but a descriptor for the object does not exist in the DBD01 table space of database DSNDB01. This error can be produced when one of the following is true:

- The Version 8 catalog has been migrated into Version 7. There is an object in the Version 7 catalog but there is not a descriptor for the object in the Version 7 directory (DBD01).
- The user did not completely restore the DBD01 table space during a recovery process.
- There is a DB2 logic error.

In the message, 'type' is either 'DATABASE' or 'SPACE' and 'name' is the eight character name of the database, table space, or index space.

System action: The process continues normally.

System programmer response: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNT602I NO SPACES FOUND

Explanation: This message is issued for multiple line WTO. It is produced when a DISPLAY DB(name) SPACENAM(...) is entered and no spaces were found to satisfy the request. SPACENAM(...) might have been entered as a range, a subset, or a list of spaces.

DSNT603I spacetype name NOT type

Explanation: This message is issued for multiple-line WTO. It is produced when no table space or index space listed in the DISPLAY DB(...) SPACENAM(...) satisfies your request.

spacetype
TABLESPACE if the space is a table space or INDEXSPACE if the space is an index space.

name The name of the space.

type A character string that indicates the type of request specified. The string is a list of the following, delimited by 'OR':

- 'ACTIVE' if the keyword ACTIVE is specified
- 'RESTRICTED' if the keyword RESTRICT is specified.

System action: Message DSNT301I or DSNT302I is produced if no object is found and the keywords ACTIVE and RESTRICT are omitted.

DSNT604I NO INFORMATION AVAILABLE

Explanation: This message is issued for multiple line WTO. It is produced when there is no LOCK or USE information available. When DISPLAY DB(name) SPACENAM(...) is entered, and SPACENAM(...) is a list of spaces, this message indicates that no table spaces or index spaces whose names are listed have LOCK or USE information available. If DISPLAY DB(name) SPACENAM(...) is entered, and SPACENAM(...) is a range or a subset of spaces, no table spaces and index spaces within the specified database have LOCK or USE information available. This message is produced only when the keywords USE and LOCKS are specified.

DSNT694I DATABASE name CANNOT BE STOPPED. IT CONTAINS THE DDL REGISTRATION TABLES

Explanation: The database specified in the message cannot be stopped because it contains the application and object registration tables. Install SYSADM or install SYSOPR authority is needed to stop the database.

System action: The database specified in the message is not stopped.

User response: Reissue the command with the proper authority.

DSNT695I TABLESPACE name CANNOT BE STOPPED. IT CONTAINS A DDL REGISTRATION TABLE

Explanation: The table space specified in the message cannot be stopped because it contains an application or object registration table. Install SYSADM or install SYSOPR authority is needed to stop the table space.

System action: The table space specified in the message is not stopped.

User response: Reissue the command with the proper authority.

DSNT696I INDEXSPACE name CANNOT BE STOPPED. IT IS DEFINED ON A DDL REGISTRATION TABLE

Explanation: The index space specified in the message cannot be stopped because it is defined on an application or object registration table. Install SYSADM or install SYSOPR authority is needed to stop the index space.

System action: The index space specified in the message is not stopped.

User response: Reissue the command with the proper authority.

DSNT700I table-name IS THE ACTIVE RESOURCE LIMIT SPECIFICATION TABLE

Explanation: This is a response message to the DISPLAY RLIMIT command. The resource limit facility (governor) is currently active and is using table 'table-name' to determine resource limits.

System action: The system continues normal processing.

User response: See Operator Response.

Operator response: To have the governor active while utilizing a different resource limit specification table, issue the START RLIMIT command specifying the ID of the desired table. If you do not want the facility to be active, issue the STOP RLIMIT command.

System programmer response: May be used to resolve questions about what table is being used.

DSNT701I RESOURCE LIMIT FACILITY IS INACTIVE

Explanation: This is a response message to the DISPLAY RLIMIT command. It results whenever the governor is not active.

System action: Normal processing continues.

User response: See Operator Response.

Operator response: If you want to activate the governor, specify the ID of the resource limit specification table to be used and issue the START RLIMIT command.

System programmer response: If the facility is already active (that is, a DSNT703 message was previously received), verify that the DSNT703 message appeared *before* the DISPLAY RLIMIT command was entered. If uncertain, reissue the DISPLAY RLIMIT command. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 5.

DSNT702I RESOURCE LIMIT FACILITY HAS BEEN STOPPED. WAS USING table-name

Explanation: This message is the normal response for a STOP RLIMIT command. Table 'table-name' was being used to determine resource limits.

System action: The governor is stopped.

System programmer response: May be used to resolve questions about what table was being used.

DSNT703I RESOURCE LIMIT FACILITY IS ALREADY INACTIVE. STOP RLIMIT COMMAND IS IGNORED

Explanation: A STOP RLIMIT command was entered but the resource limit facility was already stopped. The command is ignored.

System action: The system continues normal processing.

DSNT704I table-name HAS BEEN STARTED FOR THE RESOURCE LIMIT FACILITY

Explanation: This message is the normal response for a START RLIMIT command. Table 'table-name' is currently being used to determine resource limits.

System action: The governor has started utilizing the specified table.

System programmer response: May be used to resolve questions about what table is being used.

DSNT705I *csect-name* **DATABASE** *name* **CANNOT BE STOPPED. IT CONTAINS AN ACTIVE RESOURCE LIMIT SPECIFICATION TABLE**

Explanation: Database *name* cannot be stopped because it contains an active resource limit specification table.

System action: The database is not stopped.

Operator response: If the database must be stopped, either stop the resource limit facility (STOP RLIMIT), or switch it to use a resource limit specification table that is not in the database (START RLIMIT,ID=xx). In a DB2 data sharing environment, you must either stop the resource limit facility on all members of the DB2 data sharing group or switch it to use a resource limit specification table that is not in the database.

DSNT706I table-name DOES NOT EXIST. START RLIMIT COMMAND FAILED

Explanation: The START RLIMIT command has been entered but failed because table 'table-name' does not exist.

System action: The START RLIMIT command is ignored.

User response: See Operator Response.

Operator response: Verify that the 1 or 2 characters following DSNRLST are what was intended to be entered as the ID on the START RLIMIT command. If not, reenter the command specifying the proper ID. If the ID appears correct, contact the system programmer.

System programmer response: DB2 was unable to find the specified table in the catalog. Verify that the FULLY QUALIFIED name is the one expected. The table name is built by appending the 1 or 2 character ID specified on the START RLIMIT command to the authorization ID.DSNRLST. The authorization ID is determined from the DSNZPARMs used when starting the subsystem.

If everything appears to be in order, from a TSO terminal with the proper authorization, enter the following SQL statement: `SELECT * FROM SYSIBM.SYSTABLES WHERE NAME=DSNRLSTxx AND CREATOR=authorization ID`

Where

DSNRLSTxx
: Table-name starting with the DSNRLST

authorization ID
: Table-name up to but not including the period.

If an entry is found, save the output and if you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: Collect the following diagnostic items:
- Console output showing the failing START command
- The output of the SELECT command.

DSNT707I START RLIMIT COMMAND FAILED. COLUMN column-name IN TABLE table-name IS INCORRECTLY DEFINED. THE COLUMN DEFINITION IS attribute WITH LENGTH length AND NULL ATTRIBUTE null-attribute. THE COLUMN DEFINITION SHOULD BE attribute WITH LENGTH length AND NULL ATTRIBUTE null-attribute

Explanation: The START RLIMIT command has been entered but failed because TABLE 'table-name' does not have the proper column definition for a resource limit specification table. The current column definition is given as well as the proper column definition.

System action: The START RLIMIT command is ignored.

User response: See Operator Response.

Operator response: Verify that the 1 or 2 characters following DSNRLST are what was intended to be entered as the ID on the START RLIMIT command. If not, reenter the command specifying the proper ID. If the ID appears to be correct, contact the system programmer.

System programmer response: TABLE 'table-name' does not have the proper column definition for a resource limit specification table. Compare the current table column definition to the proper column definition given in the message.

Problem determination: Collect the console output showing the failing START command.

DSNT708I INDEX index-name DOES NOT EXIST. START RLIMIT COMMAND FAILED

Explanation: The START RLIMIT command has been entered but failed because INDEX DSNARLxx does not exist for the corresponding resource limit specification table, DSNRLSTxx.

System action: The START RLIMIT command is ignored.

User response: See Operator Response.

Operator response: Verify that the 1 or 2 characters following DSNRLST are what was intended to be entered as the ID on the START RLIMIT command. If not, reenter the command specifying the proper ID. If the ID appears to be correct, contact the system programmer.

System programmer response: The resource limit specification table DSNRLSTxx does not define the required index DSNARLxx. Create the required index and reissue the START RLIMIT command.

Problem determination: Collect the console output showing the failing START command.

DSNT709I new-table-name NOW ACTIVE. old-table-name WAS OLD RESOURCE LIMIT SPECIFICATION TABLE

Explanation: A START RLIMIT command was entered while the facility was already active. The facility remains active and switches from using the old table name to the new one.

User response: Notify the system programmer.

Operator response: Notify the system programmer.

DSNT710I table-name ALREADY ACTIVE. START LIMIT COMMAND IGNORED

Explanation: A START RLIMIT command was entered and the governor is already active utilizing the specified table. The command is ignored.

System action: The governor continues to use table 'table-name'.

User response: See Operator Response.

Operator response: Verify that the ID specified on the START RLIMIT command was the one desired. If not, reenter the command specifying the correct ID. If it is correct, no action is needed since the facility is already active and is using the desired table.

System programmer response: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: Collect the following diagnostic items listed in Appendix A, “Problem determination,” on page 659: 1, 5.

DSNT711I COLUMN column-name IS MISSING IN TABLE table-name. SHOULD BE DEFINED AS attribute WITH LENGTH length AND NULL INDICATOR OF null-attribute

Explanation: The START RLIMIT command has been entered but failed because COLUMN 'column-name' in resource limit specification table 'table-name' is not defined. The proper column definition is given in the message.

System action: The START RLIMIT command is ignored.

User response: See Operator Response.

Operator response: Verify that the 1 or 2 characters following DSNRLST are what was intended to be entered as the ID on the START RLIMIT command. If not, reenter the command specifying the proper ID. If the ID appears to be correct, contact the system programmer.

System programmer response: TABLE 'table-name' does not have COLUMN 'column-name' defined. Use ALTER TABLE to define the column as specified in the message.

Problem determination: Collect the console output showing the failing START command.

DSNT712I INDEX index-name IS NOT FOR TABLE table-name. START RLIMIT COMMAND FAILED

Explanation: The START RLIMIT command has been entered but failed because INDEX 'index-name' is not defined for resource limit specification table 'table-name'.

System action: The START RLIMIT command is ignored.

User response: See Operator Response.

Operator response: Verify that the 1 or 2 characters following DSNRLST are what was intended to be entered as the ID on the START RLIMIT command. If not, reenter the command specifying the proper ID. If the ID appears to be correct, contact the system programmer.

System programmer response: INDEX 'index-name' is not defined in resource limit specification table 'table-name'. resource limit specification table DSNRLSTxx does not define the required index DSNARLxx. The last two characters of the table name correspond to the last two characters of the index. Drop the existing DSNARLxx index and create the correct index in the resource limit specification Table.

Problem determination: Collect the console output showing the failing START command.

DSNT713I INDEX index-name IS NOT UNIQUE. START RLIMIT COMMAND FAILED

Explanation: The START RLIMIT command has been entered but failed because INDEX 'index-name' in the resource limit specification table is not defined as a UNIQUE index.

System action: The START RLIMIT command is ignored.

User response: See Operator Response.

Operator response: Verify that the 1 or 2 characters following DSNRLST are what was intended to be entered as the ID on the START RLIMIT command. If not, reenter the command specifying the proper ID. If the ID appears to be correct, contact the system programmer.

System programmer response: INDEX 'index-name' in resource limit specification table is not UNIQUE. Redefine it as unique.

Problem determination: Collect the console output showing the failing START command.

DSNT714I INDEX index-name IS INCORRECTLY DEFINED. START RLIMIT COMMAND FAILED

Explanation: The START RLIMIT command has been entered but failed because INDEX 'index-name' in resource limit specification table 'table-name' is not properly defined.

System action: The START RLIMIT command is ignored.

User response: Contact the system programmer.

Operator response: Contact the system programmer.

System programmer response: INDEX 'index-name' in the resource limit specification table does not have the proper definition. Refer to Part 5 (Volume 2) of *DB2 Administration Guide* for an example to define the resource limit specification table.

Problem determination: Collect the console output showing the failing START command.

DSNT715E ERROR reason-code READING table-name. default WILL BE RESOURCE LIMIT FOR CORRELATION-ID correlation-id, CONNECTION-ID connection-id, LUW-ID luw-id =token

Explanation: An error occurred while the resource limit specification table 'table-name' was being read. The limit is determined from the default specified in the system parameters. This default can be:

NOLIMIT
No limit applies.

NORUN
Zero time limit.

integer The limit to be used.

The reason code that defines the read error is included in the message text, as are the 'connection-ID', 'correlation-ID', and the logical-unit-of-work identifier (LUW-ID) of the thread. An LUW-ID of an '*' indicates that the thread does not have a LUW-ID. If the LUW-ID is not an '*', the value is a logical-unit-of-work-id that is assigned to the thread followed by an '=' sign and a token value, which can be used in place of logical-unit-of-work-id in any DB2 command that

accepts logical-unit-of-work-id as input. Correlation IDs beginning with '0nn', where 'nn' is a number from 01 to 28, identify system agents.

System action: Processing continues using the system-defined default.

Operator response: Notify the system programmer.

System programmer response: This is an I/O error. Investigate the problem indicated by the reason code and preceding messages. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 5.

DSNT716E AN ERROR OCCURRED ESTABLISHING A RESOURCE LIMIT FOR CORRELATION-ID correlation-id, CONNECTION-ID connection-id, LUW-ID luw-id =token. limit WILL BE USED AS THE LIMIT

Explanation: An internal error occurred while the resource limit for the specified requester was being established. The 'limit' is determined from the default specified in the system parameters. This default can be:

NOLIMIT
No limit applies.

NORUN
Zero time limit.

integer The limit to be used.

A logical-unit-of-work identifier (LUW-ID) of '*' indicates that the thread does not have a LUW-ID. If the LUW-ID is not an '*', the value is a logical-unit-of-work-id that is assigned to the thread followed by an '=' sign and a token value, which can be used in place of logical-unit-of-work-id in any DB2 command that accepts logical-unit-of-work-id as input. Correlation IDs beginning with '0nn', where 'nn' is a number from 01 to 28, identify system agents.

This message is issued by the following CSECT: DSNTCGET

System action: Processing continues using the system-defined default.

Operator response: Notify the system programmer.

System programmer response: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 5.

DSNT717I *csect-name* **TABLE SPACE** *name* **CANNOT BE STOPPED. IT CONTAINS AN ACTIVE RESOURCE LIMIT SPECIFICATION TABLE**

Explanation: Table space *name* cannot be stopped because it contains an active resource limit specification table.

System action: The table space is not stopped.

Operator response: If the table space must be stopped, either stop the resource limit facility (STOP RLIMIT), or switch it to use a resource limit specification table that is not in the table space (START RLIMIT,ID=xx). In a DB2 data sharing environment, you must either stop the resource limit facility on all members of the DB2 data sharing group or switch it to use a resource limit specification table that is not in the table space.

DSNT718I *csect-name* **INDEX SPACE** *name* **CANNOT BE STOPPED. IT IS DEFINED ON AN ACTIVE RESOURCE LIMIT SPECIFICATION TABLE**

Explanation: Index space *name* cannot be stopped because it contains an index on an active resource limit specification table.

System action: The index space is not stopped.

Operator response: If the index space must be stopped, either stop the resource limit facility (STOP RLIMIT), or switch it to use a different resource limit specification table that does not have an index in the index space (START RLIMIT,ID=xx). In a DB2 data sharing environment, you must either stop the resource limit facility on all members of the DB2 data sharing group or switch it to use a different resource limit specification table that does not have an index in the index space.

DSNT719I TABLE table-name CANNOT BE READ. START RLIMIT COMMAND FAILED

Explanation: TABLE 'table-name' cannot be accessed. This message may be preceded by related error messages that explain why the table cannot be accessed.

System action: The START RLIMIT command failed.

User response: See Operator Response.

Operator response: TABLE 'table-name' cannot currently be accessed for reading. In the case of a deadlock or timeout, reissue the START RLIMIT command.

DSNT720I **csect-name INDEX index-name NEEDS TO BE index-type**

Explanation: The START RLIMIT command was entered but failed because the named index defined for the resource limit specification table is not defined properly.

System action: The system ignores the START RLIMIT command.

User response: Contact the system programmer.

Operator response: Contact the system programmer.

System programmer response: The named index defined on the resource limit specification table does not have the proper definition.

Problem determination: Collect the console output showing the failing START command.

DSNT721I *csect-name* **DATABASE** *name* **CANNOT BE STARTED WITH ACCESS (UT). IT CONTAINS AN ACTIVE RESOURCE LIMIT SPECIFICATION TABLE**

Explanation: Database *name* cannot be started with access (UT) because it contains an active resource limit specification table.

System action: The START DATABASE command is ignored.

Operator response: If the database must be started with access (UT), either stop the resource limit facility (STOP RLIMIT), or switch it to use a resource limit specification table that is not in the database (START RLIMIT,ID=xx). In a DB2 data sharing environment, you must either stop the resource limit facility on all members of the DB2 data sharing group or switch it to use a resource limit specification table that is not in the database.

DSNT722I *csect-name* **TABLE SPACE** *name* **CANNOT BE STARTED WITH ACCESS (UT). IT CONTAINS AN ACTIVE RESOURCE LIMIT SPECIFICATION TABLE**

Explanation: Table space *name* cannot be started with access (UT) because it contains an active resource limit specification table.

System action: The START DATABASE command is ignored.

Operator response: If the table space must be started with access (UT), either stop the resource limit facility (STOP RLIMIT), or switch it to use a resource limit specification table that is not in the table space (START RLIMIT,ID=xx). In a DB2 data sharing environment, you must either stop the resource limit facility on all members of the DB2 data sharing group or switch it to use a resource limit specification table that is not in the table space.

DSNT723I *csect-name* **INDEX SPACE** *name* **CANNOT BE STARTED WITH ACCESS (UT). IT IS DEFINED ON AN ACTIVE RESOURCE LIMIT SPECIFICATION TABLE**

Explanation: Index space *name* cannot be started with access (UT) because it contains an index on an active resource limit specification table.

System action: The START DATABASE command is ignored.

Operator response: If the index space must be started with access (UT), either stop the resource limit facility (STOP RLIMIT), or switch it to use a different resource limit specification table that does not have an index in the index space (START RLIMIT,ID=xx). In a DB2 data sharing environment, you must either stop the resource limit facility on all members of the DB2 data sharing group or switch it to use a resource limit specification table that does not have an index in the index space.

DSNT724E *csect-name* **ERROR** *reason-code* **IN READING RESOURCE LIMIT SPECIFICATION TABLE** *table-name***. FOR AUTHORIZATION ID** *auth-id* **RUNNING THREAD WITH CORRELATION ID** *correlation-id***, CONNECTION ID** *connection-id***, AND LUWID** *luw-id***, USE OF** *operation-name* **IS NOT GOVERNED.**

Explanation: An error occurred reading resource limit specification table *table-name*. *auth-id* is assumed to be allowed to perform *operation-name*. *operation-name* is BIND or PARALLEL PROCESSING.

The reason code defining the read error is included in the message text as well as the connection ID, correlation ID, and logical unit of work (LUW) ID of the thread. An LUW-ID with an asterisk (*) indicates that the request originated at this site. If not an asterisk (*), the LUW-ID is assigned to the allied thread at the requesting site. Correlation IDs beginning with '0nn', where 'nn' is a number from 01 to 28, identify system agents.

System action: If the *operation-name* is BIND,
processing continues if the *auth-id* is authorized to
perform the BIND. If the *operation-name* is PARALLEL
PROCESSING, query parallelism is not disabled and
can be used if the value of the CURRENT DEGREE
special register is 'ANY'. If the *operation-name* is
PREDGOV, the active RLF (Resource Limit Facility)
assumes that the RLST table does not contain a row for
user-predicted ASUTIME, and the execution of the SQL
statement continues as normal.

User response: Notify the system programmer.

System programmer response: Investigate the problem indicated by the reason code and preceding messages. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Operator response: Notify the system programmer.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 5.

DSNT725E AN ERROR OCCURRED WHILE GOVERNING BIND OPERATIONS OF auth-id RUNNING WITH CORRELATION-ID correlation-id, CONNECTION-ID connection-id, LUW-ID luw-id. auth-id IS ALLOWED TO PERFORM BIND OPERATIONS

Explanation: An internal error occurred while determining whether or not 'auth-id' is allowed to perform bind operations. It is assumed that 'auth-id' is allowed to perform bind operations.

An LUW-ID '*' indicates that the request originated at this site. If not an '*', the LUW-ID is assigned to the allied thread at the requesting site. Correlation IDs beginning with '0nn' where 'nn' is a number from 01 to 28, identify system agents.

This message is issued by the following module: DSNTCAUT

System action: Processing occurs assuming the authorization ID is allowed to perform bind operations.

User response: Notify the system programmer.

System programmer response: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Operator response: Notify the system programmer.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 5.

DSNT726I *csect-name* **DATABASE** *name* **AND ANY SPACE IT CONTAINS CANNOT BE STARTED WITH ACCESS (UT). IT IS THE ACTIVE COMMUNICATIONS DATABASE**

Explanation: The database specified in the message cannot be started with access (UT) because it is the active communications database. In addition, spaces within the active communications database cannot be started with access (UT).

System action: The START DATABASE command is ignored.

Operator response: If the database or any space within it must be started with access (UT), you must stop the distributed data facility (STOP DDF). In a DB2 data sharing environment, you must stop the distributed data facility on all members of the DB2 data sharing group.

DSNT735I object name CANNOT BE STARTED. IT CONTAINS OR IS DEFINED ON THE DDL REGISTRATION TABLE(S)

Explanation: The object specified in the message cannot be started because it contains or is defined on the DDL registration table(s). Install SYSADM or install SYSOPR authority is needed to start the object.

In the message, 'object' is either:

- DATABASE if the specified database contains the application and object registration tables
- TABLESPACE if the specified table space contains an application or object registration table
- INDEXSPACE if the specified index space is defined on an application or object registration table

System action: The object specified in the message is not started.

User response: Reissue the command with the proper authority.

DSNT736I *csect-name* **ASYNCHRONOUS STOP DATABASE COMMAND HAS COMPLETED FOR COMMAND:** *command_string*

Explanation: Asynchronous processing for the STOP DATABASE command completed for the command whose syntax is shown in the message.

command_string is the STOP DATABASE command string for the command that is completed. The string always uses the abbreviation DB for DATABASE and SPACE for SPACENAM to save characters.

Only the first 100 characters of the STOP DATABASE command are displayed.

System action: Asynchronous processing for the STOP DATABASE command is complete.

System programmer response: No action is required.

DSNT800I *csect-name* **DB2 CANNOT IDENTIFY TO IRLM. IRLM CANNOT CONNECT TO ITS DATA SHARING GROUP. THE LOCK STRUCTURE THAT IRLM ATTEMPTED TO ACCESS WAS** *strname***.**

Explanation: IRLM failed to connect to the IRLM data sharing group. This message accompanies abend reason code 00E30806.

System action: DB2 terminates abnormally with reason code 00E30806.

System programmer response: Some possible reasons why IRLM could not connect to its data sharing group are:

- The lock structure is not properly defined in the active MVS CFRM administrative policy. The lock structure name is a concatenation of the DB2 data sharing group name and the string 'LOCK1' separated by an underscore (_). For example, if the DB2 data sharing group name is DSNCAT, DSNCAT_LOCK1 is the lock structure name.

  Ensure that the lock structure is properly defined in the active CFRM policy. The name of the lock structure that IRLM attempted to access is given in this message. IRLM issues message DXR138E to indicate that the lock structure definition could not be accessed in the active CFRM policy.
- IRLM could not connect the lock structure. The reason for the connect failure is given in IRLM message DXR135E.
- IRLM could not join its data sharing group. The reason for the join failure is given in IRLM message DXR134E.
- IRLM joined the wrong group. The set of IRLMs that are connected to a given lock structure must all be members of the same group. Ensure that the GROUP parameter is correctly specified in the IRLM startup procedure. If you give an incorrect group name, IRLM connects the lock structure, and then times out waiting for responses from other connectors. IRLM issues message DXR133I for the timeout.

DSNT801I ACTION A (ADD) DOES NOT ALLOW GENERIC DISPLAY (% IN DATA FIELD).

Explanation: You have to specify a specific data item in the DATA field.

This message is issued by the following panel: DSN8SSH

System action: The panel is redisplayed, and the cursor is placed on the field in error.

User response: Enter specific data in the DATA field.

DSNT802I ONLY ACTION D (DISPLAY) IS ALLOWED FOR DEPARTMENT STRUCTURE.

Explanation: You cannot add, update, or erase a department structure. You can only display it.

This message is issued by the following panel: DSN8SSH

System action: The panel is redisplayed, and the cursor is placed on the field in error.

User response: Change the ACTION field to D, or change the OBJECT field to DE or EM.

DSNT803I *csect-name* **DB2 CANNOT IDENTIFY TO IRLM. THE REQUESTED FUNCTION LEVEL IS NOT SUPPORTED.**

Explanation: An attempt was made to identify to an IRLM that does not support the level of function DB2 needs. This message accompanies abend reason code 00E30800.

System action: DB2 terminates abnormally with reason code 00E30800.

System programmer response: Ensure that the STEPLIB statement in your IRLM startup procedure is referencing the correct IRLM code library and that IRLM is at the appropriate level of maintenance.

DSNT804I *csect-name* **THERE ARE MODIFY LOCKS OWNED BY THIS DB2 THAT HAVE BEEN RETAINED**

Explanation: When DB2 disconnected from IRLM during DB2 shutdown, IRLM indicated that there were still outstanding modify locks. These locks were retained by IRLM. The retained locks continue to protect the resources on which they are held from requests from other DB2s in the data sharing group. That is, IRLM rejects any lock request from another DB2 that is incompatible with the retained state of the lock. The locks are retained until the terminating DB2 subsystem is restarted and the resources on which the locks are held are recovered.

System action: DB2 continues with its shutdown processing.

System programmer response: No response is necessary.

DSNT805I NAME TYPE PART STATUS LPL PAGES

Explanation: This message indicates the beginning of a set of messages used to display information as a result of a DISPLAY DATABASE command when the keyword LPL is specified. This message contains the headings of the display. (The message appears in one continuous line across the screen, not as shown above.) This message is always followed by one or more DSNT806I messages containing the column data. Refer to message DSNT806I for a description of the data these headings identify.

DSNT806I *space type part status LPL pages*

Explanation: This message contains the display information resulting from a DISPLAY DATABASE command when the keyword LPL is specified. It contains the column data describing the status of a table space or index space in the database listed in the preceding DSNT362I message. This message is always preceded by its column headings message DSNT805I. The message number for this message is not displayed.

The message fields have the following meanings:

space The table space name or index space name.

type TS for a table space, IX for an index space, LS for a LOB table space.

part The partition number. It is blank for a nonpartitioned table space or index space. For nonpartitioned secondary indexes, it is the logical partition number preceded by the character L (for example, L0001). For data-partitioned secondary indexes, the prefix for this value will be the character D (for example, D0001).

status One or more of the following, delimited by commas. The maximum length is 18 characters:

ACHKP
The table space is in aux check pending state. ACHKP on a table space indicates that CHECK DATA detected an orphan or missing LOB or an out-of-sync condition between the row in the base table and the LOB in the LOB table space.

AUXW The table space is in aux warning state. AUXW on a LOB table space indicates that one or more LOBs in the LOB table space is invalid. AUXW on a base table space indicates that CHECK DATA detected an orphan or missing LOB or an out-of-sync condition between the row in the base table and the LOB in the LOB table space.

CHKP The table space is in check pending status.

COPY An image copy is required for this object (table space or data set within this table space). STOP is also displayed. If the image copy is required because DB2 encountered a problem during RESTART, STOPE is displayed instead of STOP.

GRECP
The object (a table space, a table space partition, index space, or index space partition) is marked recovery pending because of the failure of a group buffer pool. This status value appears only on spaces. It does not apply to databases. See *DB2 Data Sharing: Planning and Administration* for information about resolving this restricted status.

LPL The object has pages or ranges of pages that are unavailable because of logical or physical damage.

LSTOP The logical partition is stopped.

RECP The table space or index space is in recovery pending status. The member must be restarted to restore full access to the data. For logical partitions, the RECP status can appear as RBDP*.

RBDP* indicates that the entire index is inaccessible to SQL applications, however only the logical partition needs to be recovered.

RO The database, table space, or index space is started for read activity only.

RW The database, table space, or index space is started for read and write activity.

STOP The database, table space, or index space is stopped.

STOPE DB2 implicitly stopped the table space or index space because of an internal inconsistency. A DSNT500I or DSNT501I message is issued at time of error detection to indicate the inconsistency.

STOPP A stop is pending or is in process for this database, table space, or index space.

UT The database, table space, or index space is started for utility processing only.

UTRO Although the table space or index space is started as RW, a utility is in process and only RO access is allowed.

UTRW The table space or index space is started as RW and a utility is in process.

UTUT Although the table space or index space is started as RW, a utility is in process and only UT access is allowed.

LPL pages
The pages in the specified *space* that are unavailable because of logical or physical damage.

DSNT807I *csect-name* **IRLM PURGE OF RETAINED LOCKS FAILED. RETURN CODE =** *retcode* **SUB-CODE =** *subcode*

Explanation: DB2 issued a PURGE request to IRLM to purge retained locks for this DB2. The PURGE request did not complete successfully. This message shows the return code and subcode from IRLM.

This message is issued by the following CSECT: DSNTRSTE

System action: DB2 abends with reason code 00E30808. Restart processing is terminated.

System programmer response: Refer to the IRLM codes.

DSNT809E *csect-name* **FEEDBACK FROM IRLM IDENTIFY INDICATED IRLM DID NOT CONNECT TO THE LOCK STRUCTURE**

Explanation: DB2 identified to IRLM, but cannot continue because IRLM did not connect to the lock structure. Global locking is ignored. This message accompanies abend reason code 00E30809.

System action: DB2 abnormally terminates with reason code 00E30809.

System programmer response: This condition happens if you are running DB2 in data sharing mode, but the associated IRLM is running with local scope (SCOPE=LOCAL in the IRLM startup procedure). If you are running DB2 in data sharing mode, ensure that the associated IRLM startup procedure specifies SCOPE=GLOBAL. Conversely, if you are not running DB2 in data sharing mode, ensure that the associated IRLM startup procedure specifies SCOPE=LOCAL.

DSNT810E *csect-name* **DB2 CANNOT IDENTIFY TO IRLM. AN INVALID LOCK STRUCTURE NAME WAS DETECTED.**

Explanation: DB2 could not identify to IRLM because of an invalid lock structure name. This message accompanies abend reason code 00E30807.

System action: DB2 startup terminates abnormally with reason code 00E30807.

System programmer response: This condition happens if you are not running DB2 in data sharing mode, but the associated IRLM is running with global scope (SCOPE=GLOBAL in the IRLM startup procedure). If you are not running DB2 in data sharing mode, ensure that the associated IRLM startup procedure specifies SCOPE=LOCAL. Conversely, if you are running DB2 in data sharing mode, ensure that the associated IRLM startup procedure specifies SCOPE=GLOBAL.

DSNT811E *csect-name* **DB2 CANNOT IDENTIFY TO IRLM. REQUEST VIOLATES ONE DB2 PER IRLM RULE.**

Explanation: DB2 could not identify to IRLM because another DBMS is already identified to the IRLM. This message accompanies abend reason code 00E30802.

System action: DB2 abends with reason code 00E30802.

System programmer response: IRLM Version 2.1 enforces a one to one (1:1) mapping between IRLM and DB2. Multiple DBMS subsystems can no longer use the same IRLM. Ensure that:

- Each DB2 subsystem is associated with its own IRLM subsystem.
- The correct IRLM procedure name and subsystem name are specified in the DB2 system parameters.

DSNT812I **–MEMBER NAME** *member-name*

Explanation: This message is issued with messages DSNT389I, DSNT394I or DSNT396I for each thread in a data sharing environment. This additional line reports the member name of the DB2 subsystem accessing this object.

If the characters '(CO)' follow the member name, then this member is the castout owner for the page set or partition. The castout owner for a page set or partition is the DB2 member responsible for doing all of the castout I/O for that page set or partition.

The message number and subsystem recognition character are not included in this message.

DSNT813I **----** *status(es) page(s)*

Explanation: This message follows message DSNT389I, DSNT392I, DSNT394I, DSNT396I or DSNT806I to display more statuses, or more pages that are in the logical page list (LPL).

LPL pages can be in two formats, single pages (represented by the page number) or a range of pages (represented by the starting page number followed by a dash and the ending page number in the range).

The message number and subsystem recognition character are not included in this message because this message is part of a DISPLAY DATABASE data line.

DSNT814I **NON-LOCAL INFORMATION UNAVAILABLE FOR** *object = name*

Explanation: One member of the DB2 data sharing group was unable to obtain information from another member for one of these reasons:

- Overflow of the maximum buffer area sent to the DB2 subsystem running the DISPLAY. This buffer area contains the DISPLAY information.
- Problems occurred while requesting DISPLAY DB information from peer DB2 subsystems.

Terms pertinent to this message:

object Either DATABASE or SPACE

name The name of the DATABASE or SPACE

User response:

- Issue DISPLAY GROUP to see if any DB2 members are down or unresponsive and then reissue the DISPLAY DATABASE command.
- Specify individual database names or space names instead of specifying '*'.

| **DSNT815I THRU** *part*

| **Explanation:** This message follows message DSNT392I
| to indicate the last in a range of partitions that share
| the same STATUS information as the partition indicated
| in the preceding message DSNT392I. This message can
| only occur when the preceding message DSNT392I
| does not contain information for the following values:

| • *to-page*
| • *hi-page*
| • *cat-name*
| • *piece*

| Refer to message DSNT392I for a description of these
| fields.

| The message number and subsystem recognition
| character are not included in this message because this
| message is part of a DISPLAY DATABASE data line.

DSNT816I *csect-name* **THE REAL-TIME**
STATISTICS COLLECTION PROCESS
IS STOPPED

Explanation: The statistics database stopped
successfully, and the real-time statistics process is
halted.

Severity: 0 (informational)

System action: DB2 will not externalize in-memory
statistics.

| **DSNT818I** *csect-name* **INVALID SPACENAME**
| **SPECIFIED**

| **Explanation:** Incorrect syntax is specified for the table
| space name while issuing START/STOP/DISPLAY
| Database Commands.

| **Severity:** 0 (informational)

| **System action:** DB2 will not process the command.

Chapter 18. DSNU... messages

The severity values shown for the utility messages that follow are returned as the job-step condition code from the job step during which the message is issued. If additional messages having higher severity values are issued during the same job step, the higher value is reflected as the job-step condition code.

Severity values have the following meanings:

0	Informational	No error occurred
4	Warning	A condition was detected; the user might need to take further action
8	Error	An error was detected and processing could not continue

DSNU000I *csect-name* **OUTPUT START FOR UTILITY, UTILID =** *uuuuuuu*

Explanation: This message marks the start of output for the given utility job. The utility ID used to identify the utility job is 'uuuuuuu'.

Severity: 0 (informational)

System action: Utility processing continues.

DSNU001I *csect-name*- **RELEASE DEPENDENT MODULE** *module-name* **WAS NOT FOUND**

Explanation: The system could not find the specified module in the indicated library (private library, job library, or link library).

System action: The utility job is abended.

User response: Ensure that the load module library is indicated correctly and that the indicated library contains the requested program. If the indicated library does not contain the requested program, contact the system programmer.

System programmer response: The last three characters of the module name indicate the DB2 release level of the subsystem identified in the parameters passed to DSNUTILB. However, LOAD could not find the release dependent utility module for that release in the indicated library. This is probably a maintenance problem.

DSNU002I *csect-name* **- COULD NOT LOAD RELEASE DEPENDENT MODULE** *module-name* **CODE=code REASON=reason**

Explanation: During the load of 'module-name', MVS returned the indicated system completion code and reason.

System action: The utility job is abended.

User response: Refer to the appropriate MVS publication.

DSNU003I *csect-name* **NOT INVOKED APF AUTHORIZED**

Explanation: The stored procedure was not invoked authorized.

Severity: 8 (error)

System action: The utility is not run by the stored procedure.

User response: Notify the system programmer.

System programmer response: Check the following items:

- The DSNUTILS or DSNUTILU entry in SYSIBM.SYSPROCEDURES specifies that the stored procedure runs in a WLM-established address space (WLM_ENV not blank)
- The DSNUTILS or DSNUTILU load module is link edited with AC(1)
- The DSNUTILS or DSNUTILU load module is in an APF authorized library
- All the libraries in STEPLIB JOBLIB are APF authorized

If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 532, 46, 60, 61.

© Copyright IBM Corp. 1983, 2008

DSNU004I *csect-name* **UTILITY utility NOT SUPPORTED**

Explanation: Refer to *DB2 Utility Guide and Reference* for a description of utilities supported by the stored procedure.

Severity: 8 (error)

System action: The utility is not run by the stored procedure.

User response: Run the utility as a batch job.

DSNU005I *csect-name input record*

Explanation: This message displays the utility statements as they were provided as input to the utility parser. An additional error message follows this message and identifies why the utility statement caused a parser error.

Severity: 8 (error)

System action: Utility processing is not initiated. No SYSIBM.SYSUTIL record is created.

User response: Correct the input error, and resubmit the job. Do not restart the job.

DSNU006I *csect-name* **DB2 SYSTEM** *system-name* **HAS NOT BEEN STARTED**

Explanation: Utility startup processing has been unable to connect to the indicated DB2 subsystem, because a -START DB2 command has not been issued for that subsystem.

Severity: 8 (error)

System action: Utility processing is terminated without invoking the requested utility or utilities.

User response: Correct your JCL to specify the correct subsystem-name on the EXEC statement, and resubmit the job. Otherwise, contact the system programmer to determine when the subsystem will be available for processing.

Operator response: Contact the system programmer to determine if a -START DB2 command should be issued for the indicated subsystem.

System programmer response: Have the operator issue a -START DB2 command for the indicated subsystem.

Problem determination: DB2 must be started in order to execute DB2 utilities.

DSNU007I *csect-name* **DB2 SYSTEM** *system-name* **DOES NOT EXIST**

Explanation: Utility startup processing has been unable to connect to the indicated DB2 subsystem, because it does not exist on this system.

Severity: 8 (error)

System action: Utility processing is terminated without invoking the requested utility or utilities.

User response: Correct your JCL to specify the correct subsystem-name on the EXEC statement, and resubmit the job. Contact your system programmer if you are not sure which DB2 subsystem you should specify.

System programmer response: Ensure that the correct DB2 subsystem has been started via the -START DB2 command.

Problem determination: DB2 must be started in order to execute DB2 utilities.

DSNU008I *csect-name* **SPECIFIED USER FAILED VALIDITY CHECK**

Explanation: Utility startup processing has been unable to connect to DB2, because the user failed the connection validity check.

Severity: 8 (error)

System action: Utility processing is terminated without invoking the requested utility or utilities.

User response: Contact the system programmer for assistance with your installation's security checking.

System programmer response: Locally defined RACF or equivalent security system connection authorization have denied the user access to DB2. Permit DB2 connection authorization if this user should be allowed access.

DSNU009I *csect-name* **SPECIFIED USER NOT AUTHORIZED FOR DB2 UTILITIES**

Explanation: Utility startup processing for the user indicated on the job statement failed, because the user is not authorized to execute DB2 utilities.

Severity: 8 (error)

System action: Utility processing is terminated without invoking the requested utility or utilities.

User response: Contact your system programmer for assistance with your installation's security checking.

System programmer response: If the USER specified something on the job statement that is incorrect, correct it, and resubmit the job. Use the SQL GRANT command to give the user the desired authority.

DSNU010I *csect-name* **UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE =** *rr*

Explanation: The utility completed processing with a return code of 'rr'. This indicates that no terminating error was detected.

Severity: 0 (informational) or 4 (warning)

System action: In most cases, the SYSIBM.SYSUTIL record for this job step is deleted from SYSIBM.SYSUTIL. However, if the utility used was REORG and UNLOAD(PAUSE), the SYSIBM.SYSUTIL record remains intact so that REORG can be restarted at the RELOAD phase.

User response: If the return code ('rr') is 0, no response is required.

If the return code ('rr') is 4, the utility output should be examined for other messages.

DSNU012I *csect-name* **UTILITY EXECUTION TERMINATED, HIGHEST RETURN CODE =** *rr*

Explanation: The utility execution was terminated. An error was detected that would not allow processing to continue. The return code has a value of 8.

Severity: 8 (error)

System action: Utility processing is terminated.

User response: Examine utility output for other error messages.

DSNU013I *csect-name* **JOB WITH DUPLICATE UTILID FOUND IN SYSUTIL**

Explanation: An entry with the same qualifier was found in the SYSIBM.SYSUTIL table. The current job did not specify restart.

System action: The utility job is abended with reason code X'00E40018'.

User response: If the job was intended to restart the utility, RESTART should be added to the EXEC parameters. If the job was intended to rerun the utility, the -TERM utility command should be used to delete the entry that has the same qualifier from the SYSIBM.SYSUTIL table. If the job had no relation to the SYSIBM.SYSUTIL entry job, a new qualifier should be used.

DSNU014I *csect-name* **JOB WITH DUPLICATE UTILID CURRENTLY EXECUTING**

Explanation: A job with the same qualifier was found to be executing.

System action: Utility processing is terminated.

User response: If the job with the same qualifier that is currently executing is correct, either wait for the job to complete before using the same qualifier, or use another qualifier. If the job currently executing is incorrect, the -TERM command can be used to terminate it, and the failing job can be resubmitted. Do not -TERM UTILITY a utility job step that is currently active in REORG beyond the UNLOAD phase.

DSNU015I *csect-name* **JOB WITH UTILID NOT FOUND IN SYSUTIL FOR RESTART**

Explanation: A utility job was submitted to restart a utility job that had previously failed. The SYSIBM.SYSUTIL entry for the specified qualifier provided was not found.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Use the -DISPLAY utility command to determine the correct qualifier. The job can then be resubmitted.

DSNU016I *csect-name* **UTILITY BATCH MEMORY EXECUTION ABENDED, REASON=X0***rrr cause*

Explanation: The utility batch address space has abended. If X'0rrr' is four characters in length (that is, X'00C1'), the abend was a system X'rrr' abend. If X'0rrr' is four characters in length and system abend X'000' was indicated, a user abend was issued. If X'0rrr' is eight characters in length, the abend was a system X'04E' abend, and X'0rrr' is an abend reason code.

If the value of general register 2 is not zero, then *cause* will contain CAUSE=X'*cccccccc*' where X'*cccccccc*' is the value of general register 2. This information is for service purposes only.

System action: The utility job is abended. The SYSIBM.SYSUTIL entry for the job is retained in the SYSUTIL table.

User response: Consult the system programmer to obtain information on failure analysis. If the cause of the failure is correctable, the utility job can be restarted. If the error cannot be corrected, the -TERM command should be used to delete the failing job's utility id from SYSIBM.SYSUTIL.

System programmer response: If a system abend was encountered (4 characters), refer to the appropriate MVS publication. If an abend reason code was encountered (8 characters), refer to *DB2 Utility Guide and Reference* for problem determination.

DSNU017I *csect-name* **UTILITY DATA BASE SERVICES MEMORY EXECUTION ABENDED, REASON=X0***rrr*

Explanation: The database services address space has abended. If X'0rrr' is four hexadecimal digits in length (for example, X'00C1'), the abend was an MVS X'rrr' abend. If X'0rrr' is eight hexadecimal digits in length, the abend was a DB2 subsystem X'04E' abend, and X'0rrr' is an abend reason code.

If the value of general register 2 is not zero, then *cause* will contain CAUSE=X'*cccccccc*' where X'*cccccccc*' is the value of general register 2. This information is for service purposes only.

System action: The utility job is abended. The entry for the job is retained in the SYSIBM.SYSUTIL table.

User response: Consult with the system programmer to obtain information about failure analysis. If the cause of the failure is correctable, the utility job can be restarted. If the error cannot be corrected, the -TERM command should be used to delete the failing job's utility id from SYSIBM.SYSUTIL.

System programmer response: If an MVS system abend was encountered (4 hexadecimal digits), refer to the appropriate MVS publication. If an abend reason code was encountered (8 hexadecimal digits), look up the reason code in *DB2 Codes*, and if you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNU018I *csect-name* **JOB IS NOT RESTARTABLE DUE TO -START ACCESS FORCE COMMAND**

Explanation: A utility job was submitted to restart a utility job that previously failed. Although the SYSIBM.SYSUTIL entry for the specified qualifier was found, the job is not restartable because of a -START DATABASE ACCESS FORCE command. The -START DATABASE ACCESS FORCE command deleted the utility-in-process state of one or more (but not all) target objects in the job.

System action: The utility job is abended with reason code X'00E40018'.

User response: If the specified qualifier is correct, use the -TERM UTILITY command to terminate the utility and resubmit the job as a nonrestartable job. If the -TERM UTILITY command fails, use the -START DATABASE ACCESS FORCE command to force ALL target objects (table spaces and indexes) in the job. When the last object is forced, the SYSIBM.SYSUTIL record is deleted. If the qualifier is incorrect, correct it and restart the job.

DSNU019I *csect-name* **THIS UTILITY IS NOT ALLOWED AT A DB2 TRACKER SITE**

Explanation: When a DB2 subsystem serves as a tracker site, it cannot run utility *csect-name*. A DB2 subsystem is also a tracker site if the value of subsystem parameter TRKRSITE is YES. You may run only the following utilities at a tracker site DB2:

- REPORT
- RECOVER without the TORBA, TOLOGPOINT, or TOCOPY options
- DIAGNOSE

Severity: 8 (error)

System action: The utility terminates.

System programmer response: If this DB2 subsystem does not need to serve as a remote site tracker system, remove its tracker site status before re-trying the utility. You may remove the tracker site status by stopping the DB2 subsystem, then restarting it with TRKRSITE = NO.

User response: Contact the system programmer.

DSNU020I *csect-name* **JOB STEP INCOMPLETE**

Explanation: The REORG utility was invoked with the UNLOAD PAUSE option. This causes the immediate suspension of the job after the unload phase completes. Subsequent utilities that are invoked in the same job step are not run.

Severity: 4 (warning)

System action: Further utility processing is terminated.

User response: Resubmit the job with the RESTART(PHASE) parameter after the desired user action has been taken with the unload data set.

Problem determination: Refer to *DB2 Utility Guide and Reference* for a description of the REORG utility and the UNLOAD PAUSE option.

DSNU021I *utility-name* **UTILITY WITH UTILID =**
utilid **WHICH ENDED IN PHASE =**
phase-name, **IS ATTEMPTING TO**
RESTART (*restart-type***)**

Explanation: This message indicates that the specified
utility, *utility-name*, with utility id = *utilid* is attempting
to restart using RESTART(*restart-type*). One of the
following restart-types was specified:

• *restart-type* PHASE from the beginning of the phase
that was being processed.

• *restart-type* CURRENT from the last commit that was
taken during the execution of the utility phase.

The type of restart might be changed during restart
execution depending on the utility and the phase where
the original job ended, as specified in *DB2 Utility Guide*
and Reference.

Severity: 0 (Informational)

System action: The system continues processing.

User response: Resubmit the job with the
RESTART(PHASE) parameter after the desired user
action has been taken with the unload data set.

Problem determination: Refer to *DB2 Utility Guide and*
Reference for a description of the REORG utility and the
UNLOAD PAUSE option.

DSNU024I *csect-name* **PARM FIELD ERROR** *parm-field*

Explanation: An error was detected in the PARM field on the EXEC statement. 'parm-field' was the input parameter field.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Correct the indicated parameter on the EXEC DD statement, and resubmit the job.

Problem determination: Refer to *DB2 Utility Guide and Reference* for parameter requirements.

DSNU025I *csect-name* **INVALID RESTART PARAMETER (restart)**

Explanation: Only the following values may be specified for the *restart* parameter: *NO* or *null*, CURRENT, or PHASE.

Severity: 8 (error)

System action: The utility is not run by the stored procedure.

User response: Correct and re-run.

DSNU028I *csect-name* **UNABLE TO CONNECT TO DB2, REASON=X***reason code*

Explanation: A connection to the DB2 subsystem failed. 'reason-code' is the DB2 reason code that was returned.

Severity: 8 (error)

System action: The utility job is terminated.

User response: Check the subsystem ID specified in the EXEC DD statement. If it is correct, the subsystem was not ready for processing. Ask the system programmer when the requested subsystem will again be available for processing.

DSNU030I *csect-name* **UNABLE TO ALLOCATE** *dsn*, **RC=***rr*, **CODE=***ccc*

Explanation: The utility was unable to allocate a necessary image copy data set. *dsn* is the data set name that the utility attempted to dynamically allocate. *rr* is the return code from the MVS dynamic allocation service, and *ccc* is the reason code associated with the return code.

System action: The utility proceeds along one or the following paths:

If the data set was an incremental image copy data set, the utility proceeds as if it and any later incremental image copies did not exist.

If the data set was a full image copy data set, the utility attempts to use the most recent prior full image copy data set.

If the data set is used by the COPYTOCOPY utility, and the utility job step includes an OPTIONS control statement with the EVENT(ITEMERROR,SKIP) option, COPYTOCOPY processes the next item in the list.

The allocation failure may create other problems that will cause the utility to fail. This is indicated by other messages.

User response: If the job fails, contact the system programmer.

System programmer response: If the utility fails because of the allocation error (as shown by a subsequent message), correct the error, then resubmit the job.

Problem determination: Refer to the appropriate MVS publication for a complete description of the possible return and reason codes from dynamic allocation.

DSNU031I **UNABLE TO** *action dsn*, **RC=***rc*,
CODE=*rsn*

Explanation: A utility was unable to perform the
indicated action on a data set. Possible actions are
UNALLOCATE or CATALOG. The data set name that
the utility was processing is *dsn*. The return code from
the MVS DYNALLOC (SVC 99) or CATALOG (SVC 26)
service is *rc*. The reason code associated with the return
code is *rsn*.

System action: The system action following this
message varies. Following catalog errors from both
DYNALLOC and CATALOG, processing continues.
Following DYNALLOC errors unrelated to the catalog,
the utility job is abended with reason code X'00E40076'.

Problem determination: For CATALOG errors, refer to the "Return Codes from Catalog" section of *DFSMSdfp Advanced Services* for possible return codes. For DYNALLOC errors, refer to *MVS/ESA Programming: Authorized Assembler Services Guide* for possible return and reason codes.

DSNU032E **I/O ERROR ON WORKFILE,** *xxxxxx*

Explanation: An I/O error was encountered on a necessary OS/VS data set. 'xxxxxx' is a message that describes the error and the data set in error. This message is returned to the DB2 utility by SVC 68 (SYNADAF) from BSAM. The data set may have been needed for an unload data set, image copy data set, or load or reorganization work file data set. The utility is stopped at the beginning of the phase that used the OS/VS data set.

System action: The utility job is abended. The utility job step is in the stopped state.

Operator response: Except in the case of an unload data set during the RELOAD phase, the job can be

restarted if an undamaged output data set is provided.

Problem determination: Refer to the appropriate MVS publication for the format of the error message.

DSNU033I *csect-name* **THE CONTROL STATEMENT DOES NOT IDENTIFY THE OBJECT ON WHICH THE UTILITY OPERATES**

Explanation: In the utility control statement, a keyword or operand was specified before the name of the table space, index, index space, or list on which the utility operates.

Severity: 8 (error)

System action: Parsing is discontinued and utility processing is terminated.

User response: Specify the object of the utility before any keywords or operands that relate to the object.

Problem determination: Refer to *DB2 Utility Guide and Reference* for correct utility syntax.

DSNU034I **DATA SET TYPE INVALID FOR DDNAME 'dddddddd'**

Explanation: A required DD statement with DDNAME='dddddddd' was found to have a data set type which is not valid for this application.

Severity: 8 (error)

System action: The utility is terminated with a return code 8.

User response: The data set must be capable of storing intermediate data that can be retrieved by DB2. Check that the data set is not specified as type DUMMY, SYSIN, or SYSOUT. If you require assistance in specifying a valid data set type, consult the system programmer.

System programmer response: Correct the JCL and resubmit the job.

DSNU035I *csect-name* **UNABLE TO FIND DD CARD FOR DDNAME, 'dddddddd'**

Explanation: A DD statement with ddname = 'dddddddd' was expected, but not found.

Severity: 8 (error)

System action: The utility is terminated with a return code 8.

User response: If the DDNAME is one you submitted, review and correct it. If the DDNAME was not provided by you, but by the system, consult the system programmer.

System programmer response: Correct the JCL and resubmit the job.

DSNU036I *csect-name* **UNABLE TO OPEN DCB FOR DDNAME** *dddddddd*

Explanation: The DCB associated with DDNAME 'dddddddd' could not be opened.

System action: The utility job is abended with a reason code of X'00E40070'. The utility job step is in the stopped state.

User response: If the ddname is one you submitted, review and correct it. If the ddname was not provided by you but by the system, consult the system programmer.

System programmer response: Correct the job control language and resubmit the job.

DSNU037I *csect-name* **- RECFM=U FOR DDNAME** *dddddddd*

Explanation: A data set to be used by the utility has an undefined record format RECFM=U. All data sets must be fixed or variable, blocked or unblocked.

System action: The utility job is abended with a reason code of X'00E40070'. The utility job step is in the stopped state.

Operator response: Notify the system programmer.

System programmer response: Ensure that no data sets involved in the utility job have RECFM=U. Resubmit the job after ending the stopped utility by the TERM UTILITY command, or do a RESTART(PHASE) of the stopped utility.

DSNU038I *csect-name* **BLOCK SIZE FOR DCB, DDNAME** *dddddddd*, **IS TOO SMALL, MUST BE AT LEAST** *bbbbb* **BYTES**

Explanation: The data set with DDNAME 'dddddddd' has a block size that is too small to accommodate a record that needs to be written.

System action: The utility job is abended with a reason code of X'00E40079'. The utility job step is left in the stopped state.

User response: Delete the data set in error. Change the JCL so that the block size for the data set with DDNAME 'dddddddd' is at least 'bbbbb' bytes. Resubmit the utility job with the RESTART(PHASE) option.

DSNU039I *csect-name* **- LRECL=0 FOR DDNAME** *dddddddd*

Explanation: A data set to be used by the DB2 utility has a record length of zero after OPEN. All data sets with a record format of fixed or fixed blocked must have a valid record length greater than zero.

System action: The utility job is abended with reason

code X'00E40070'. The utility job step is left in the stopped state.

User response: If the DDNAME is for a data set that you submitted, review and correct it. It might be necessary to provide a valid LRECL in the DCB parameter in the JCL. If the DDNAME was not provided by you but by the system, consult the system programmer.

Operator response: Notify the user or system programmer.

System programmer response: Correct the data set definition, or add DCB information to the JCL. Do a RESTART(PHASE) of the stopped utility, or resubmit the job after issuing a -TERM UTILITY command for the stopped utility.

DSNU040I *csect-name* **ERROR ACCESSING DDNAME** *ddname*, **DECB=Xhex string**

Explanation: An error has occurred during BSAM access of a utility temporary file identified by 'ddname'. The DECB indicating the error is given as a hexadecimal string.

System action: The executing utility issues an X'04E' abend with abend reason code '00E40071' following this message.

User response: Refer to the information given in the Problem Determination section of this message. Correct the problem. Either issue the -TERM command to terminate the utility job and resubmit it, or specify the RESTART parameter and resubmit it.

Operator response: An operator with SYSOPR authority can issue the -TERM UTILITY command to release the resources claimed by the stopped utility (this may not be necessary).

Problem determination: The DECB provided contains the ECB in the first four bytes. The first byte of the ECB contains status bits that indicate the cause of the error. Refer to the appropriate MVS publication for the meaning of the status bits.

DSNU041I *csect-name* **INVALID CONCATENATED DATA SET FOR DDNAME** *dddddddd*

Explanation: A non-input concatenated data set was detected. Only input data sets can be concatenated.

System action: The utility job abended with reason code X'00E40070'. The utility job step is in the stopped state.

Operator response: Notify the system programmer.

System programmer response: Remove the concatenated data sets from the data set with DDNAME 'dddddddd'. Resubmit the job after ending the stopped utility with the TERM UTILITY command, or restart the utility at the beginning of the stopped phase by using RESTART(PHASE).

DSNU042I *csect-name* **SORT PHASE STATISTICS. NUMBER OF RECORDS=***n* **ELAPSED TIME=***hh/mm/ss*

Explanation: This message provides statistics on the SORT phase of a utility. The number of records sorted, 'n', and the duration of the SORT phase in hours, minutes, and seconds are provided.

Severity: 0 (informational)

System action: The system continues to process normally.

DSNU044I *csect-name* **ERROR FROM SORT COMPONENT RC=***n*, **UTILITY STOPPED**

Explanation: A nonzero return code 'n' was returned from an MVS sort. Error messages from sort are printed on the UTPRINT data set.

System action: The executing utility job is placed in the stopped state.

User response: Determine the cause of the error. If the error can be corrected, the utility may be restarted from the beginning of the sort phase by using the RESTART(PHASE) option. If the error is such that the utility cannot complete, the -TERM command should be issued to remove the stopped utility from the system. The -TERM utility option should rarely be exercised when the stopped utility is REORG.

Operator response: An operator with SYSOPR authority can issue the -TERM UTILITY command to release the resources claimed by the stopped utility (this may not be necessary).

Problem determination: This abend is accompanied by abend reason code X'00E40005', and an SVC dump is requested. In most cases, error messages from MVS sort identify the problem and the dump is not needed. If sort is unable to open the UTPRINT data set, only the message and the abend are provided. For example, if the UTPRINT DD statement is omitted, correct the problem with UTPRINT, and resubmit the job.

DSNU046I *csect-name* **UTILITY STATEMENT IS TOO LONG OR TOO COMPLEX**

Explanation: The utility command and associated control statements have caused internal fields to exceed DB2 subsystem limits.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Reduce the size or complexity of the utility statement where possible. For example, a LOAD into several tables may be broken up into a separate load statement for each table or the size of comparison

fields for 'when', 'continueif', and 'nullif' operands may be reduced in size.

Problem determination: Refer to *DB2 Utility Guide and Reference* for correct operand definitions.

DSNU047I *csect-name* **A REQUIRED DD CARD OR TEMPLATE IS MISSING. NAME=***missing-dd*

Explanation: The DB2 utility being executed requires a data set to continue processing. The DDNAME or TEMPLATE name for the missing data set is provided.

Severity: 8 (error)

System action: Utility processing terminates.

Operator response: Notify the system programmer.

System programmer response: Correct the JCL to include the indicated DD statement or add a TEMPLATE control statement. Refer to *DB2 Utility Guide and Reference* for information about required DD statements.

DSNU048I *csect-name* **INVALID OPERAND FOR KEYWORD** *kkkkkkkk*

Explanation: The utility parser has determined that the operand specified for keyword 'kkkkkkkk' is invalid.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Correct the operand in error, and resubmit the utility job. Refer to *DB2 Utility Guide and Reference* for correct operand definitions.

DSNU049I *csect-name* **- INVALID OPERAND** *oooooooo* **FOR KEYWORD** *kkkkkkkk*

Explanation: The utility parser has determined that operand 'oooooooo' cannot be specified for keyword 'kkkkkkkk'. This includes a negative value following the keywords MAXRO or DELAY.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Correct the operand in error, and resubmit the utility job. Refer to *DB2 Utility Guide and Reference* for proper option specifications.

DSNU050I *csect-name utility-statement*

Explanation: This message echoes the utility statement that was input from SYSIN.

Severity: 0 (informational)

System action: Utility processing continues.

DSNU051I *csect-name* **INSUFFICIENT OPERANDS FOR KEYWORD** *kkkkkkkk*

Explanation: An operand that is required when the keyword 'kkkkkkkk' is specified was omitted.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Correct the operand specification, and resubmit the job. Refer to *DB2 Utility Guide and Reference* to determine the correct set of operands for 'kkkkkkkk'.

DSNU052I *csect-name* **KEYWORD kkk INCOMPATIBLE WITH OTHER KEYWORDS OR OPERANDS**

Explanation: The specified keyword 'kkk' cannot be used with other keywords or operands.

Severity: 8 (error)

System action: Processing terminates.

User response: Refer to the *DB2 Utility Guide and Reference* for the right syntax for the specified keyword.

DSNU053I *csect-name* **FIELD** *cccccccc* **NOT FOUND**

Explanation: Either the column name specified, 'cccccccc', was not found in the catalog as a valid column name for the table being processed, or a WHEN keyword field name was not found in the field name list.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Correct the LOAD utility statement, and resubmit the job.

Problem determination: Valid column names can be found in the SYSIBM.SYSCOLUMNS table.

DSNU054I *csect-name* **TABLESPACE** *ssssssss* **NOT FOUND**

Explanation: The operand of the TABLESPACE keyword, 'ssssssss', did not specify an existing table space name.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Correct the table space name and/or database name, and resubmit the job.

Problem determination: Valid table space names can be found in the SYSIBM.SYSTABLESPACE catalog table.

DSNU055I *csect-name* **INDEX** *iiiiiiii* **NOT FOUND**

Explanation: The index name specified, 'iiiiiiii', was invalid.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Correct the index name, and resubmit the job.

Problem determination: Valid index names can be found in the SYSIBM.SYSINDEXES catalog table.

DSNU056I *csect-name* **TABLE** *ttttttt* **NOT FOUND**

Explanation: A utility specified a table name '*ttttttt*', but the table does not exist.

Severity: 8 (error)

System action: The error is sufficient to terminate further processing of the utility job step. The utility job step is terminated, and resources are released to their prior state.

User response: Correct the table name and/or database name, and resubmit the utility job step.

Problem determination: The correct table names can be found by querying the SYSIBM.SYSTABLES catalog table.

DSNU057I *csect-name* **DATABASE** *database-name* **NOT FOUND**

Explanation: The operand of the DATABASE keyword, 'database-name', did not specify an existing database name.

Severity: 8

System action: The error is sufficient to terminate further processing of the utility job step. The utility job step is terminated, and resources are released to their prior state.

User response: Correct the database name and resubmit the utility job step.

Problem determination: The correct database names can be found by querying the SYSIBM.SYSDATABASE catalog table.

DSNU058I *csect-name utility-name* **UTILITY NOT ALLOWED AGAINST AN OBJECT IN RECOVER PENDING STATE**

Explanation: An attempt has been made to execute a utility against an object that is in RECOVER PENDING STATE. CHECK, COPY, MODIFY, LOAD, QUIESCE, REORG, and RUNSTAT are not allowed.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Remove the object from the RECOVERY PENDING STATE by using either LOAD REPLACE, RECOVER, or REPAIR against the object. Resubmit the job.

System programmer response: Determine which of the three utilities, LOAD REPLACE, RECOVER, or REPAIR, should be used to remove the object from the RECOVER PENDING STATE.

DSNU059I *csect-name utility-name* **UTILITY NOT ALLOWED TO RUN ON A ROSHARE READ DATABASE**

Explanation: An attempt was made to execute a utility on a ROSHARE READ database. Only the RUNSTATS, DIAGNOSE, and REPAIR DBD utilities are allowed to run on read-only shared databases.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Resubmit the utility job on the owning system, if necessary.

DSNU060I *csect-name* **USER** *userid* **NOT AUTHORIZED FOR** *util-name* **UTILITY ON DATABASE** *dbase-name*

Explanation: User *userid* attempted to execute utility *util-name* against an object in database *dbase-name*. The user was not authorized to execute utilities for database *dbase-name*, the user's authorization did not include this utility, or the user's authorization did not have a valid seclabel.

Severity: 12 (severe error)

System action: Utility processing is not initiated.

User response: Ensure that the correct database was specified. If this operation involves a table with Multilevel security with row level granularity, then this user needs to have a valid security label to execute this utility.

System programmer response: Determine why the user was denied access. Either the authorization should be corrected, or another authorized user should invoke the utility.

Problem determination: The following catalog tables can be examined to determine who has proper authority: SYSIBM.SYSUSERAUTH, SYSIBM.SYSDBAUTH, or SYSIBM.SYSRESAUTH.

DSNU061I *csect-name* **USER** *userid* **NOT AUTHORIZED, UTILID=***utilid*

Explanation: The named user issued a utility command and was not authorized to do so.

The -TERM UTILITY command can only be issued by:

- The submitter of the utility job.

- Users with SYSOPR, SYSADM, or SYSCTRL authority.
- Users with DBADM, DBCTRL, or DBMAINT authority over every database used by the utility.

  Even for users with sufficient database authority, the TERM UTILITY command will fail in these situations:
  - The user issued the command before the utility determined what databases it would access.
  - The utility job included the DIAGNOSE, REPORT, or STOSPACE utility. Those utilities must be terminated by the submitter of the job or by a holder of SYSOPR, SYSCTRL, or SYSADM authority.

Any user can issue a -DISPLAY UTILITY command.

System action: The -TERM UTILITY request for utility identifier *utilid* is rejected.

User response: Take the appropriate action:

- Wait to allow the utility to identify the databases it will access; then reissue the command.
- Obtain the required authorization and reissue the command.
- Have another user with proper authorization issue the command.

DSNU062I *csect-name util-name* **UTILITY MAY NOT BE RUN ON OBJECT** *qualifier.object-name*

Explanation: An attempt was made to execute utility 'util-name' against object 'qualifier.object-name'. This is not allowed. The named object cannot be processed by the named DB2 utility.

Severity: 8 (error)

System action: The utility job is terminated.

User response: Remove the invalid utility control statements and resubmit the job.

DSNU063I *csect-name utility-name* **UTILITY NOT ALLOWED AGAINST AN OBJECT IN REBUILD PENDING STATE**

Explanation: An attempt has been made to execute a utility against an object that is in REBUILD PENDING STATE. CHECK, COPY, MODIFY, LOAD, QUIESCE, REORG, and RUNSTAT are not allowed.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Remove the object from the REBUILD PENDING STATE by using either LOAD REPLACE, RECOVER, or REPAIR against the object. Resubmit the job.

System programmer response: Rebuild the index with the REBUILD utility to remove the object from REBUILD PENDING state.

DSNU064I *csect-name* **UTILITY NOT ALLOWED AGAINST SYSTEM DATABASE**

Explanation: An attempt was made to execute a utility against database DSNDB01 or DSNDB06. The following utilities are not allowed against DSNDB01:

LOAD
REORG TABLESPACE against SYSUTILX (allowed against other DSNDB01 table spaces)
CHECK DATA
REPAIR DBD
RUNSTATS
MODIFY STATISTICS

The following utilities are not allowed against DSNDB06:

LOAD (except SYSIBM.SYSSTRINGS)
CHECK DATA
REPAIR DBD

Severity: 8 (error)

System action: Execution of the utility job is stopped.

User response: Remove the invalid utility control statements and resubmit the job.

DSNU065I *csect-name* **UTILITY NOT ALLOWED AGAINST A WORK FILE**

Explanation: An attempt has been made to execute a utility against database DSNDB07. Utilities are not allowed against work files.

Severity: 8 (error)

System action: Execution of the utility job is stopped.

User response: Remove the invalid utility control statement(s) and resubmit the job.

DSNU066I *csect-name* **OPERAND** *oooooo* **NOT ALLOWED FOR** *uuuuuu* **UTILITY against** *tttttt* **TABLE**

Explanation: An attempt was made to specify the REPLACE option when executing the LOAD utility on a DB2 catalog table. It is not permissible to replace existing rows in a DB2 catalog table. Rows can only be added to the existing rows in a DB2 catalog table.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Correct the operand, and resubmit the job.

DSNU067I *csect-name* **- TABLE SPACE** *table-space-name* **IS NOT A LOB TABLE SPACE**

Explanation: The table space specified, *table-space-name*, is not a LOB table space. The object of the utility must be LOB table space.

System action: Utility returns completion code=8.

Programmer response: Correct the utility statement.

DSNU068I *csect-name* **-** *function* **IS NOT SUPPORTED FOR LOB TABLE SPACES**

Explanation: The function specified, *function*, is not supported for the LOB table space specified. The LOB table space reported by the message cannot be explicitly named in the utility statement.

Severity: 8 (error)

System action: Utility execution is terminated.

User response: Correct the utility statement.

DSNU069I *csect-name* **- AUXILIARY TABLE** *table-name* **NOT PERMITTED AS EXCEPTION TABLE**

Explanation: A auxiliary table was specified as an exception table.

System action: Utility returns completion code=8.

Programmer response: Specify an exception table that is not an auxiliary table.

DSNU070I *csect-name* **KEYWORD OR OPERAND** *x* **INVALID WITH** *y*

Explanation: The use of KEYWORD or OPERAND 'x' is inconsistent with KEYWORD or OPERAND 'y'.

Severity: 8 (error)

System action: The control statement in error is rejected and processing is halted.

User response: Correct the control statement in error and resubmit the job.

DSNU071I *csect-name* **KEYWORD** *x* **REQUIRES KEYWORD** *y*

Explanation: When keyword 'x' is specified, keyword 'y' must also be specified.

Severity: 8 (error)

System action: The control statement in error is rejected and processing is halted.

User response: Correct the control statement in error, and resubmit the job.

DSNU072I *csect-name* **INVALID KEYWORD(S) SPECIFIED FOR TABLESPACE LIST**

Explanation: The specified keyword cannot be included in a list of table spaces.

Severity: 8 (error)

System action: Processing terminates.

User response: Remove the specified keyword from the list of table spaces.

DSNU073I *csect-name* **- KEYWORD 'wwwww' IGNORED**

Explanation: The keyword, 'wwwww' is no longer supported.

Severity: 0 (informational)

System action: Utility processing proceeds.

Problem determination: Refer to *DB2 Utility Guide and Reference* for the current syntax.

DSNU074I *csect-name* **-** *utility* **UTILITY NOT SUPPORTED FOR LOB TABLE SPACE:** *table-space-name*

Explanation: The utility statement failed because the table space specified was a LOB space and the utility does not support LOB table spaces.

System action: The statement cannot be executed.

Programmer response: Correct the utility statement.

DSNU075I *csect-name* **- TABLE** *table_name*, **LOB COLUMN** *column-name* **- DEFINITION IS NOT COMPLETE**

Explanation: The definition of the LOB column, *column-name* in table *table_name* is not complete.

System action: Utility returns completion code=8.

Programmer response: Complete the LOB column definitions. A LOB column definition is not complete until the LOB table space, auxiliary table and index on the auxiliary table have been created for the LOB column.

DSNU076I *csect-name* **- KEYWORD=***keyword* **INVALID FOR** *type* **TABLE SPACE=***table_space_name*

Explanation: The keyword, *keyword*, is invalid when the object of the utility is a *type* table space.

System action: Utility returns completion code=8.

Programmer response: Correct the utility statement.

DSNU077I *csect-name* - *utility* **IS PROHIBITED BECAUSE** *object* **IS** *pending-state*

Explanation: The *utility* specified is prohibited because the *object* is in a *pending-state*.

Severity: 8 (error)

System action: Utility terminates and returns completion code=8.

Programmer response:

- Run the CHECK LOB utility if the object is a LOB table space and the state is CHECK PENDING.
- Run the REBUILD INDEX utility if the object is an auxiliary index and the state is REBUILD PENDING.
- Run the CHECK INDEX utility if the object is an auxiliary index and the state is CHECK PENDING.

DSNU078I *csect-name* **INVALID DBCS IDENTIFIER** *iiii*

Explanation: The DBCS identifier iiii is invalid. Reason codes are:

101 An odd number of bytes exists between the shift-out and the shift-in

102 No shift-in was found at the end of the identifier

103 DBCS blanks X'4040' are not allowed

104 There are no characters between the shift-out and the shift-in

105 Shift-out can not be the first byte of the DBCS character between the shift-out and the shift-in.

Severity: 8 (error)

System action: Processing is terminated.

User response: Correct the DBCS identifier and resubmit the job.

Problem determination: Refer to Chapter 2 of *DB2 SQL Reference* to determine the correct syntax for DBCS identifiers.

DSNU079I *csect-name* **UNBALANCED PARENTHESES IN CONTROL STATEMENT**

Explanation: The utility parser has determined that there are unbalanced parentheses in the input string.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Determine the location of the missing parenthesis and insert it. The utility job can then be resubmitted.

Problem determination: Refer to *DB2 Utility Guide and Reference* to determine the correct syntax for option specification.

DSNU080I *csect-name* **NO UTILITY STATEMENTS FOUND IN** *dd-name*

Explanation: The data set identified by dd-name did not contain any text. The system might have processed the end of file prematurely, or this job requires that the utility statements be skipped.

Severity: 4 (warning)

System action: Processing continues.

User response: If the data set is SYSIN, provide at least one utility statement, if none now exists, or verify that this job executes no utility statements. If the data set is the LISTDEF (SYSLISTD) or TEMPLATE (SYSTEMPL) data set, provide the required control statements in that data set or in SYSIN.

DSNU081I *csect-name* **OPERAND 'oooooooo' EXCEEDS KEYWORD 'kkkkkkkk' MAXIMUM LENGTH**

Explanation: The utility parser has determined that operand 'oooooooo' exceeds the maximum length for keyword 'kkkkkkkk'.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Correct the operand, and resubmit the job.

Problem determination: Refer to *DB2 Utility Guide and Reference* to determine the correct operand specification.

DSNU082I *csect-name* **INVALID KEYWORD** *kkkkkkkk*

Explanation: The given keyword was not recognized within the context of this control statement.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Correct the keyword, and resubmit the job.

DSNU083I *csect-name* **INVALID HEXADECIMAL OPERAND** *hhhhhhhh*

Explanation: The utility parser has determined that the operand *hhhhhhhh* is invalid as a hexadecimal operand. It contains one of the following errors:

- the operand contains a character that is not hexadecimal
- it has an odd number of characters
- for a UX or GX string constant, it consists of a number of characters that is not a multiple of 4

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Correct the keyword, and resubmit the job.

DSNU084I *csect-name* **INVALID GRAPHIC OPERAND** *gggg*

Explanation: The utility parser has determined that the operand 'gggg' is invalid as a hexadecimal operand. The graphic operand is either missing one or both of the shift characters or is not an even number of characters.

Severity: 8 (error)

System action: Processing is terminated.

User response: Correct the utility statement, and resubmit the job.

DSNU085I *csect-name* **INVALID SPECIAL CHARACTER OPERAND** *ssss*

Explanation: The utility parser has determined that the operand 'ssss' is invalid as a special character operand. For a special character operand to be invalid, it must have either a shift-out or a shift-in character and be missing the corresponding shift character, or not have an even number of characters between the shift-out, shift-in characters.

Severity: 8 (error)

System action: Processing is terminated.

User response: Correct the utility statement, and resubmit the job.

DSNU086I *csect-name ttttt* **I/O ERRORS ON SPACE=** *dddddddd***. DATA SET NUMBER=** *nnn***. I/O ERROR PAGE RANGE=** *aaaaaa, bbbbbb***.**

Explanation: DB2 detected an error. This message is issued for, but not limited to the following reasons:

- I/O errors were detected on the named object. *ttttt* is either 'READ' or 'WRITE'. *nnn*; is the data set number. *aaaaaa* is the first damaged page and *bbbbbb* is the last damaged page of the data set.
- A broken page was detected.
- A page parity error was detected for a 'must complete' agent.
- A read or write I/O error was detected. A LPL recovery might be in progress so the page is added to the WEPR instead of the LPL.

Severity: 0 (informational)

System action: Processing continues, but the pages within the error range are no longer accessible.

System programmer response: Determine the appropriate level of recovery, and invoke the recovery utility. If user-defined data sets are being used and full data set or table space RECOVER is to be used, then, prior to invoking the RECOVER utility, the table space must be stopped, new data set(s) defined, and the table space restarted. If the data set is part of a ROSHARE READ database, determine whether it was a link or DASD failure. It there was a link failure, then fix the link, and issue a -START DATABASE with the ACCESS(FORCE) parameter. If the error was caused by a DASD failure, then correct the problem from the system on which the database was defined as ROSHARE OWNER.

Problem determination: Refer to Part 4 (Volume 1) of *DB2 Administration Guide* for a discussion of database recovery procedures.

There may be an IOS message on the MVS console SYSLOG indicating the type of I/O error that occurred. If an IOS message was issued, follow the Problem Determination directions for the message.

DSNU087I *csect-name* **INVALID OPERAND TYPE FOR KEYWORD** *kkkkkkkk - oooooooo*

Explanation: The utility parser has determined that the operand 'oooooooo' is the wrong type for the keyword 'kkkkkkkk'. An example of such an error would be a decimal operand where only a hexadecimal operand is allowed.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Correct the keyword, and resubmit the job.

Problem determination: Refer to *DB2 Utility Guide and Reference* to determine valid operand types for the designated keyword.

DSNU088I *csect-name* **INVALID CHARACTER OPERAND** *cccccccc*

Explanation: The utility parser has determined that the operand 'cccccccc' is invalid as a character operand.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Correct the keyword, and resubmit the job.

Problem determination: Refer to *DB2 Utility Guide and Reference* for correct specification of the character operand.

DSNU089I *csect-name* **OPERAND** *oooooooo* **IS TOO LONG**

Explanation: The utility parser has determined that the operand 'oooooooo' is too long for the keyword specified.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Correct the keyword, and resubmit the job.

Problem determination: Refer to *DB2 Utility Guide and Reference* for correct operand length specification.

DSNU090I *csect-name* **DD OR TEMPLATE NAME** *nnn* **IS INVALID FOR KEYWORD** *kkk*

Explanation: The DDNAME or TEMPLATE *nnn* was used as an improper input of the specified keyword *kkk* or from one of the reserved ddnames: SYSIN, SYSPRINT, UTPRINT, SORTWKnn, SORTLIB.

Severity: 8 (error)

System action: Processing terminates.

User response: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNU091I *csect-name* **DDNAME** *nnn* **IS A DUPLICATE**

Explanation: The DDNAME 'nnn' was used as operand in the input statement more than once.

Severity: 8 (error)

System action: Processing terminates.

User response: Change the ddname to a different one.

DSNU092I *csect-name* **NUMBER OF OPERANDS FOR KEYWORD** *kkk* **EXCEEDS MAXIMUM**

Explanation: The keyword 'kkk' contains more than the maximum number of operands.

Severity: 8 (error)

System action: Processing terminates.

User response: Refer to *DB2 Utility Guide and Reference* for the limit on the number of operands for the specified keyword.

DSNU093I *csect* **- INVALID CHARACTER STRING** *string* **IN CONTROL STATEMENT**

Explanation: An invalid character string *string* was found in a utility control statement where a keyword, operand, operator or punctuation was expected.

Severity: 8 (error)

System action: Processing terminates.

User response: Correct the utility control statement, and resubmit the job.

DSNU094I *csect-name* **KEYWORD 'wwwww' IS DUPLICATED**

Explanation: The keyword 'wwwww' is not allowed twice.

Severity: 8

System action: The control statement in error is rejected and the processing is halted.

User response: Correct the control statement in error, and resubmit the job.

DSNU095I *csect-name* **UTILITY** *uuuuuuuu* **INCONSISTENT WITH LAST INVOCATION**

Explanation: This message can occur when a utility job step is restarted and a utility statement was modified, added or deleted from SYSIN.

Severity: 8 (error)

System action: Utility processing is terminated. The utility job step remains in SYSIBM.SYSUTIL in the stopped state.

User response: Restart the utility job with the statements in the same order as they were in when the job was first run. If a statement was modified, change the statement back to the format in the original invocation and restart the job. Utility statements should not be altered for restart.

DSNU096I *csect-name* **OPERAND** *oooooooo* **INCONSISTENT WITH LAST INVOCATION**

Explanation: When the utility job step was restarted, the indicated operand was found to be inconsistent with the restart point.

System action: The restart of utility processing is not initiated.

User response: To restart, restore the operand to its prior definition. If this is not acceptable, terminate the utility job step, take corrective action, and resubmit the job.

Operator response: An operator with SYSOPR authority can terminate the utility job step (this may not be necessary). A REORG utility that is beyond the UNLOAD phase should rarely be terminated.

Problem determination: Refer to *DB2 Utility Guide and Reference* for operand definitions. Refer to Part 4 (Volume 1) of *DB2 Administration Guide* for a discussion of utility restart constraints.

DSNU097I *csect-name - input* **WILL BE REQUIRED SYNTAX IN FUTURE RELEASES**

Explanation: The identified 'input' is not required in this release, but it will be part of the required syntax in future releases.

Severity: 4 (warning)

System action: Utility processing continues.

Programmer response: Refer to *DB2 Utility Guide and Reference* for the current syntax.

DSNU099I *csect-name* **- IRLM '***req-type***' REQUEST FAILED, IRLM RETURN CODE = X'***return-code***' IRLM REASON CODE = X'***reason-code***'**

Explanation: The request to the IRLM to obtain or release a lock failed for the reasons specified in *return-code* and *reason-code*. In a non data-sharing environment, the requested lock is the utility serialization lock. In a data sharing environment, the requested lock is the utility serialization lock, the utility identifier lock, or the utility execution lock. The IRLM return and reason codes are documented in *DB2 Codes*. The type of request is indicated by *req-type*.

System action: Utility and command processing terminates.

User response: Refer to the IRLM messages and codes section for problem determination and the appropriate action. Restart the utility by specifying the RESTART parameter, or terminate the utility with the -TERM command and then resubmit the job. If this message was received during the -DIS UTIL, -TERM UTIL, or -START DATABASE ACCESS(FORCE) command, then the command can be reissued.

System programmer response: Determine the cause of the IRLM error and take appropriate action.

Problem determination: Obtain the system console log output, dump, and SYS1.LOGREC information. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNU100I *csect-name* **USERID =** *userid* **MEMBER =** *member-name* **UTILID =** *utilid* **PROCESSING UTILITY STATEMENT** *statement-number* **UTILITY =** *utility-name* **PHASE =** *utility-phase* **COUNT =** *object-count* **NUMBER OF OBJECTS IN LIST =** *list-size* **LAST OBJECT STARTED =** *last-started* **STATUS = STOPPED**

Explanation: This message is issued in response to a DISPLAY UTILITY command. The utility job with identifier *utilid* stopped during processing. The utility stopped for one of these reasons:

- The REORG utility stopped in response to its UNLOAD(PAUSE) option. The utility is stopped in the RELOAD phase.
- The utility is stopped because of a DB2 subsystem failure or cancellation (for example, a STOP DB2 command).

In a data sharing environment, *member-name* identifies the DB2 member. In a non data-sharing environment, *member-name* is blank. At the time of the request, *object-count* records or pages were processed.

If the utility job step is restarted with the RESTART(PHASE) option, it begins with PHASE *utility-phase*. However, not all utilities have phases. Those that do have phases restart from the beginning of that phase.

COUNT indicates the number of pages or records processed in a utility phase. COUNT has different meanings for different utilities. For utilities not mentioned below, ignore this field.

- For CHECK INDEX, LOAD, REBUILD INDEX, and REORG: number of records processed (log records for the LOG phase of REORG), or 0 (for the SWITCH or BUILD2 phase of REORG).
- For COPY, MERGECOPY, RECOVER (restore phase), and RUNSTATS: number of pages processed.
- For STOSPACE: number of table spaces or indexes processed.

If the utility is processing a list of objects, *list-size* gives the total number of objects in the list and *last-started* gives the item number of the last object for which processing was started. These values may not be accurate in the UTILINIT before the list size has been determined.

Severity: 0 (informational)

System action: DB2 waits for directions to either restart or terminate the utility.

User response: If you have the required authority, restart or terminate the utility job step.

To restart the job step, you must have authority to run the indicated utility and to complete any other utility functions in the utility job step.

To terminate the utility with the -TERM UTILITY command, you must be the originator or have SYSOPR or SYSADM authority.

Operator response: If necessary, and if you have SYSOPR authority, you can issue the -TERM UTILITY command to release the resources claimed by the stopped utility.

DSNU103I *csect-name* **- UNABLE TO ACCESS SYSUTIL TABLE AT THIS TIME**

Explanation: An -ALTER UTILITY, -DISPLAY UTILITY, or -TERM UTILITY command was issued at the same time that the DSNDB01.SYSUTILX table space was the object of an active utility job. The utility job might be holding locks that would cause a deadlock condition to occur.

Severity: 8 (error)

System action: The command processing is terminated.

User response: Enter the command again. This message is issued in response to each command until the utility action on the DSNDB01.SYSUTILX table space completes.

DSNU104I *csect-name* **- DSNDB01.SYSUTILX IN A PENDING STATE PROHIBITS FURTHER PROCESSING**

Explanation: The -ALTER UTILITY, -TERM UTILITY, or -DISPLAY UTILITY command was issued when the DSNDB01.SYSUTILX table space was in a pending state, which prohibits further processing. The pending state was either copy or recovery pending.

Severity: 0 (informational)

System action: Command processing is terminated.

User response: Notify the system programmer so that the pending condition can be resolved.

System programmer response: Issue a -DISPLAY DATABASE command to determine the state of the DSNDB01.SYSUTILX table space. If the state is copy pending, do a full image copy. If the state is recovery pending, use the RECOVER utility to recover DSNDB01.SYSUTILX. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNU105I *csect-name* **- USERID=***userid* **MEMBER=***member-name* **UTILID=***utilid* **PROCESSING UTILITY STATEMENT** *n* **UTILITY=***utility-name* **PHASE=***phase-name* **COUNT=***n* **NUMBER OF OBJECTS IN LIST=***list-size* **LAST OBJECT STARTED=***last-started* **STATUS=ACTIVE**

Explanation: This message is issued in response to a DISPLAY UTILITY command.

It is also issued by the REORG utility if the user specified SHRLEVEL REFERENCE or CHANGE and DB2 *estimates* that the switch phase will not start by the deadline specified by the DEADLINE phrase.

This message gives the user an estimate of how much processing the utility completed. The *utility-name* utility in the utility job step identified by *utilid* was executing in phase *utility-phase* at the time the -DISPLAY UTILITY request was made. In a data sharing environment, *member-name* identifies the DB2 member. In a non-data-sharing environment, *member-name* is blank.

COUNT *n* is the number of pages or records processed in a utility phase. COUNT has different meanings for different utilities. For utilities not mentioned below, ignore this field.

- For CHECK INDEX, LOAD, REBUILD INDEX, and REORG: number of records processed (log records for the LOG phase of REORG), or 0 (for the SWITCH or BUILD2 phase of REORG).
- For COPY, MERGECOPY, RECOVER(restore phase), and RUNSTATS: number of pages processed
- For STOSPACE: number of table spaces or indexes processed.

If the utility is processing a list of objects, *list-size* gives the total number of objects in the list and *last-started* gives the item number of the last object on which processing was started. These values may not be accurate in the UTILINIT before the list size has been determined.

Severity: 0 (informational)

System action: The message reflects the state of DB2 at the time of the -DISPLAY UTILITY command or the time when DB2 estimated that the switch phase in the REORG utility with SHRLEVEL REFERENCE or CHANGE would not start by the deadline specified by the DEADLINE phrase.

If this message was issued for -DISPLAY UTILITY, the system continues processing. A subsequent -DISPLAY UTILITY command reflects that processing.

DSNU106I *csect-name utility-name* **UTILITY, MEMBER =** *member-name* **UTILID =** *utilid* **TERMINATING**

Explanation: This message is issued in response to a DISPLAY UTILITY command. The *utility-name* utility in the utility job step identified by *utilid* was in the process of terminating at the time the DISPLAY UTILITY request was issued. The utility is terminating in response to a previous TERM UTILITY request.

In a data sharing environment, *member-name* identifies the DB2 member. In a non data-sharing environment, *member-name* is blank.

Severity: 0 (informational)

System action: The system is stopping the utility and releasing all resources associated with the utility. After the utility terminates, DB2 can reuse the utility identifier.

DSNU107I *csect-name* **MEMBER NAME** *member-name* **IS NOT A MEMBER OF THIS DATA SHARING GROUP.**

Explanation: This message is issued by the DISPLAY UTILITY command when an invalid *member-name* is encountered in the member list.

Severity: 4

System action: The DISPLAY UTILITY command ignores the invalid *member-name* and continues to process the next member name specified.

User response: Correct the member name and reissue the command.

DSNU108I *csect-name* **DATA SHARING ENVIRONMENT IS NOT ACTIVE, MEMBER KEYWORD NOT APPLICABLE.**

Explanation: This message is issued by the DISPLAY UTILITY command when the MEMBER keyword is specified, but data sharing is not active.

Severity: 4

System action: The DISPLAY UTILITY command ignores the MEMBER keyword and its associated parameters, but continues processing.

User response: Make sure data sharing is active before issuing the DISPLAY UTILITY command with the MEMBER keyword.

DSNU109I *csect-name* **UTILITY** *utilid* **CAN ONLY BE TERMINATED OR ALTERED BY MEMBER =** *member-name*

Explanation: The -TERM UTILITY or -ALTER UTILITY command cannot terminate or alter the utility job with identifier *utilid*, because it belongs to member *member-name* of the DB2 data sharing group. Activeutility jobs can be terminated or altered only from the system on which they were running.

Severity: 8 (error)

System action: If this message was issued for the -TERM UTILITY command, the command ignores the *utilid* and continues to process the next utilid, if any. If this message was issued for the -ALTER UTILITY command, the command is rejected, and further processing of the command is ended.

User response: Make sure the *utilid* identifies the utility you want to terminate or alter. If it does, issue the command from the DB2 member identified in the message.

DSNU110I *csect-name* **UTILID =** *uuuu* **INVALID**

Explanation: This message is issued in response to an invalid specification of a UTILID on an -ALTER UTILITY, -DISPLAY UTILITY, or -TERM UTILITY command.A UTILID is a 1- to 16-byte alphanumeric token that can contain any number of periods.

Severity: 8 (error)

System action: The -ALTER UTILITY, -DISPLAY UTILITY, or -TERM UTILITY command is rejected. Further processing of the command is ended.

User response: Review the UTILID provided in the message to determine why it does not conform to the syntax rules of a utility identifier token, and reissue the request.

DSNU111I *csect-name* **- SUBPHASE=***subphase-name* **COUNT=***n*

Explanation: This message is issued only in response
to a -DISPLAY UTILITY command that is entered on
the same member where the utility is executing. It
immediately follows a prior DSNU105I message if the
utility is using subtasks to perform processing in
parallel. One of these messages exists for each subtask.

This message gives the user an estimate of how much processing the utility subtask has completed.

SUBPHASE *subphase-name* identifies the activity that the subtask was performing at the time the -DISPLAY UTILITY command was issued.

COUNT *n* is the number of records processed by the utility subtask.

Severity: 0 (informational)

System action: The system continues processing.

DSNU112I *csect-name* **- NO AUTHORIZED UTILITY FOUND FOR UTILID =** *utilid*

Explanation: This message is issued in response to an -ALTER UTILITY, -DISPLAY UTILITY, or -TERM UTILITY command. DB2 has no record of the utility job identified by the utility identifier *utilid*. The *utilid* might not be correct, or the utility might have already terminated.

Severity: 0 (informational)

System action: The system rejects the -ALTER UTILITY, -DISPLAY UTILITY, or -TERM UTILITY request for the indicated utility job.

User response: Verify that the utility identifier of the utility job to be altered, displayed, or terminated was correctly entered.To view all currently valid identifiers, issue the -DISPLAY UTILITY (*) or (partial-utility-id*) command.

DSNU113I *csect-name* **DISPLAY OR TERM UTILITY MESSAGE LIMIT HAS BEEN REACHED**

Explanation: This message indicates that the DISPLAY or TERM utility command that was specified generated more messages than will fit in the amount of storage that is available.

Severity: 0

System action: The DISPLAY or TERM UTILITY command terminates.

User response: Specify a set of utility jobs using a partial-utility-id to limit the number of utilities being displayed or terminated. See the DB2 Command Reference for more information.

DSNU114I *csect-name* **UTILITY COMMANDS SUBSYSTEM FAILURE, REASON=X***code*

Explanation: This message indicates an abend has occurred in the utility commands subsystem.

System action: The command abends. In the case of the -TERM command, termination actions taken previously under this command are backed out.

User response: Prior messages and/or the SVC dump identify the failure. If the condition can be corrected, the command may be reissued.

Problem determination: In most cases, the failure is not user related. In some cases, prior messages indicate a user error that may be corrected. The problem may be as simple as SYSIBM.SYSUTIL being temporarily locked by another command or a utility job. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNU115I *csect-name utility-name* **UTILITY WITH UTILID =** *utilid* **CAN ONLY BE TERMINATED BY A MEMBER WITH THE SAME RELEASE =** *release*

Explanation: Command TERM UTILITY cannot terminate the utility job with identifier *utilid* because it belongs to a member with a release level identified in the message. Some utilities can be terminated only from a member of the DB2 data sharing group with the same release level.

Severity: 0 (informational)

System action: Command TERM UTILITY is rejected for the indicated utility job.

User response: Make sure the *utilid* identifies the utility you want to terminate. If it does, issue command TERM UTILITY from a member with a release level identified in the message in the DB2 data sharing group.

DSNU118I *csect-name* **MAXIMUM NUMBER OF ERROR MESSAGES ISSUED**

Explanation: The utility issued the maximum number of messages that can be generated with the available memory for message buffers. DB2 suppresses any additional error messages.

Severity: 0 (informational)

System action: Utility processing continues.

User response: Refer to the message information for any other error messages that are issued by the utility and take the specified action to resolve the problems. Then rerun the utility to ensure that all errors are resolved.

DSNU119I *csect-name* **NUMBER OF OBJECTS PROCESSED BY THE UTILITY EXCEEDED THE MAXIMUM ALLOWED**

Explanation: The number of objects being processed by the utility has exceeded the maximum allowed, which includes table, table space partitions, index, index space partitions and other database objects modified or referenced by the utility. This generally happens when a utility is run on a partitioned table space with many partitions and indexes defined.

System action: The utility job is abended with reason code X'00E400A1'.

User response: Reduce the number of partitions and indexes being processed by the utility. If the utility must be run against the entire table space, you can possibly drop one or more indexes to bypass this problem.

DSNU160I *csect-name - utility-name* **UTILITY, UTILID =** *utilid* **REQUESTED TO TERMINATE, PHASE =** *utility-phase* **WAS ACTIVE**

Explanation: This message is issued in response to a -TERM UTILITY command. It acknowledges that the request for termination was received and verified. At the time the -TERM UTILITY request was received, utility 'utility-name' was ACTIVE in the utility job step identified by UTILID 'utilid'.

System action: DB2 has accepted the request to terminate the indicated utility job step. The utility terminates at its next cleanup point.

User response: This message is an acknowledgement that the request has been accepted and is in progress.

DSNU161I *csect-name* - *utility-name* **UTILITY WITH UTILID =** *util-id* **CAN NOT BE TERMINATED AT THE DB2 TRACKER SITE**

Explanation: DB2 does not permit the termination of certain utilities at a tracker site because terminating these utilities may leave the target objects in an unpredictable state. The next recovery cycle should cause the utilities to resolve. You may not terminate the following utilities at a tracker site:

- COPY
- LOAD
- REORG
- REPAIR

Severity: 8 (error)

System action: The utility does not terminate.

System programmer response: Monitor the status of these utilities after the next tracker site recovery cycle.

User response: Contact the system programmer.

DSNU162I *csect-name* - *utility-name* **UTILITY WITH UTILID =** *util-id* **IS TERMINATING WITH OBJECTS THAT HAVE BEEN STARTED WITH ACCESS(FORCE) AND THEN DROPPED**

Explanation: Database objects that are used by the terminating utility were started with the command START DATABASE ACCESS(FORCE) and then dropped.

Severity: 4 (warning)

System action: Termination process continues.

User response: If the terminating utility is Online Reorg, TERM UTILITY does not rename and delete residual DB2-managed or user-defined data sets. Those datasets must be deleted manually.

DSNU164I *csect-name cccccccc* **UTILITY, UTILID =** *uuuu* **ALREADY TERMINATING**

Explanation: This message is in response to a -TERM UTILITY command. It indicates that the utility job step indicated by UTILID 'uuuuu' has already been requested to terminate and is in the process of doing so now.

Severity: 0 (informational)

System action: DB2 is currently in the process of terminating the indicated utility job step. Upon completion, all resources will have been released to their prior state, and all tracking of the indicated UTILID will have been removed. At that point, the UTILID is available for reuse.

User response: This message indicates that either the request was redundant, or someone else is also interested in the termination of the subject utility job step.

DSNU166I *csect-name utility-name* **UTILITY, UTILID =** *uid* **NOT EXECUTING, CLEANUP COMPLETE**

Explanation: This message is in response to a -TERM UTILITY command. The 'utility-name' utility identified by UTILID 'uid' was found to be stopped at the time of the request to terminate.

Severity: 0 (informational)

System action: Cleanup is performed for the indicated stopped utility. All resources held by the utility are freed and all storage of the utility job is deleted. The utility job is no longer restartable.

System programmer response: Resources held by this utility may have prevented applications, commands, and other utilities from executing. These have been freed, so that the inhibited tasks may now be performed.

DSNU170I *csect-name* **UTILITY JOB TERMINATED BY TERM COMMAND**

Explanation: A -TERM UTILITY command was issued for this utility job step.

Severity: 8 (error)

System action: The executing utility terminates processing and frees all held resources. All storage of the utility job has been deleted. Therefore, the job is not restartable. Subsequent utilities in the same job step are not executed.

User response: All or portions of the utility job step can be resubmitted as needed.

DSNU172I *csect-name* **UTILITY JOB STOPPED BY STOP DB2 COMMAND**

Explanation: A -STOP DB2 command was issued during execution of the utility job step.

Severity: 8 (error)

System action: The executing utility stops processing but continues to hold resources. Subsequent utilities in the same job step are not executed. The storage of the utility job is retained. Therefore, the utility job step is restartable.

User response: Following a -START DB2 command, the utility job step may be restarted. Alternatively, the utility may be terminated using the -TERM UTILITY command and resubmitted as a new job. A REORG utility that has passed the UNLOAD phase should rarely be terminated.

DSNU180I *csect-name* **UTILITY IS NOT COMPATIBLE WITH THE** *utility-name* **UTILITY UTILID =** *uid* **OBJECT =** *spacename.object-name*

Explanation: Certain utilities cannot process the same table or index space at the same time. The message identifies the objects for which the incompatibility occurred.

These objects might be target objects of the utilities, a mapping table, or they might be catalog or directory table spaces that are updated by the utilities. For example, DSNDB06.SYSCOPY is updated by LOAD.

Severity: 8 (error)

System action: The starting utility is not initiated, and further utility processing in that utility job step is terminated.

User response: Use -DISPLAY UTILITY with the provided 'uid' to determine when the utility that caused your utility to be rejected is completed. Refer to Part 5 (Volume 2) of *DB2 Administration Guide* for a discussion of which utilities are mutually compatible.

DSNU181I *csect-name* **RESTART NOT ALLOWED FOR** *utility-name* **UTILITY**

Explanation: An attempt was made to restart the named utility. DB2 does not allow this utility to be restarted.

System action: Utility processing is abended. Abend reason code 00E40018 might be issued.

User response: Issue the -TERM UTIL command to terminate the utility job and then rerun the job without the RESTART parameter.

System programmer response: If necessary, take actions to prevent access to the table space before terminating the utility.

DSNU182I *csect-name utility-name* **MUST BE THE ONLY UTILITY IN SYSIN**

Explanation: The submitted utility control statements contain a utility on the table space DSNDB01.SYSUTILX. This table space is integral to controlling the execution of utilities. Control over previous or subsequent utilities in the job step would be lost if the utility were allowed to execute. Therefore, the identified utility must be the only utility in a job step.

This message is also issued if you recover the SYSUTILX indexes, DSNLUX01 and DSNLUX02, or table space DSNDB01.DBD01 and include other utility control statements in the same job step.

Additionally, this message will be issued if you invoke either the BACKUP SYSTEM or RESTORE SYSTEM utilities and include additional commands.

Severity: 8 (error)

System action: Work performed by utilities in the same utility job step prior to the starting utility persist. The utility identified and subsequent utilities are not allowed to execute, and the utility job step is placed in the stopped state.

User response: Terminate the utility execution that received this message using the -TERM UTILITY command. Submit a subsequent job step containing only the identified utility. Execute additional utilities, if any, in a separate job step.

DSNU183I *csect-name - utility-name* **UTILITY NOT COMPATIBLE WITH OTHER ACTIVE UTILITIES**

Explanation: The utility identified by 'utility-name' is executing against table space DSNDB01.SYSUTILX. This table space is integral to controlling the execution of utilities. Control over other utilities running at the same time would be lost if the utility were allowed to execute. Therefore, the identified utility must be executed alone. There must be no other active DB2 utilities in the system.

Severity: 8 (error)

System action: Utility execution is terminated.

User response: Rerun the utility when there are no other utilities executing. Use the -DISPLAY UTILITY command to see what utilities are ACTIVE. Utilities in the STOPPED state are allowed.

DSNU184I *csect-name* **- DO NOT RECOVER OBJECT qual.obj-nm UNTIL THE NEXT TRACKER SITE RECOVERY CYCLE**

Explanation: If DB2 detects an inconsistency relating to a utility's status on object 'qual.obj-nm' at the tracker site, DB2 issues this message. This event can occur if the SYSUTIL record for the utility either does not exist, or reflects incorrect information about hte status of the utility.

Severity: 8 (error)

System action: The utility issues ABEND04E with return code 00E40009.

User response: Contact the system programmer.

System programmer response: Wait for the next recovery cycle to reover the object named in this message. If the object named in this message is a catalog or directory object, you cannot recover this object, or any other objects in the list of catalog and drectory objects, or any user data or indexes. Wait until the next recovery cycle to recover the object.

DSNU185I *csect-name* **THE VSAM DATA SET DOES NOT EXIST FOR** *obj-type obj-qual.obj-name part part-number*

Explanation: The VSAM data set(s) for the specified object has not been defined yet because the object was created with DEFINE NO.

Severity: 8 (error)

System action: The utility skips the object that was created with DEFINE NO, and processing continues.

DSNU200I *csect-name* **- ERROR RANGE ON TABLE SPACE** *database.tablespace-name* **PROHIBITS PROCESSING**

Explanation: The utility encountered an error range on the identified table space or partition, which prohibits it from continuing.

Severity: 8 (error)

System action: The utility fails serialization and terminates.

User response: Correct the error range for the table space by running RECOVER ERROR RANGE. Then resubmit the original failing utility.

DSNU201I *csect-name* **- ERROR RANGE ON INDEX** *creator.index-name* **PROHIBITS PROCESSING**

Explanation: The utility encountered an error range on the identified index space or partition which prohibits it from continuing.

Severity: 8 (error)

System action: The utility fails serialization and terminates.

User response: Identify the incorrect error range by issuing the DISPLAY command for the index. Run the RECOVER or REBUILD INDEX utility to restore the index or index partition that has the error range. Then resubmit the original failing utility.

DSNU202I *csect-name* **- RECOVER PENDING ON TABLE SPACE** *database.tablespace-name* **PROHIBITS PROCESSING**

Explanation: An attempt was made to execute a utility against a table space that is in a recover pending state.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Remove the table space from the recover pending state by using either LOAD REPLACE, RECOVER, or REPAIR against the object. LOAD REPLACE with KEEPDICTIONARY is not allowed on a table space (with compression active) in the recover pending state. Resubmit the job.

System programmer response: Determine which of the three utilities, LOAD REPLACE, RECOVER, or REPAIR, should be used to remove the object from the recover pending state.

DSNU203I *csect-name* **- RECOVER PENDING ON INDEX** *creator.index-name* **PROHIBITS PROCESSING**

Explanation: An attempt was made to execute a utility against an index that is in a rebuild pending state.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Remove the index from the rebuild pending state by using the REBUILD INDEX utility against the index. . Resubmit the job.

DSNU204I *csect-name* **- PAGESET REBUILD PENDING ON INDEX** *creator.name* **PROHIBITS PROCESSING**

Explanation: An attempt was made to execute a utility against a logical partition of an index when the index is in PAGESET rebuild pending state,

Severity: 8 (error)

System action: Utility processing terminates.

User response: Remove the index space from PAGESET rebuild pending state by running the REBUILD INDEX utility against the entire index, not just a single logical partition. Resubmit the job.

DSNU205I *csect-name* **- LOGICAL ERROR RANGE (LOGICAL PAGE LIST) ON TABLE SPACE** *database.tablespace-name* **PROHIBITS PROCESSING**

Explanation: The utility cannot continue because it encountered a logical error range (logical page list) on the identified table space or partition.

Severity: 8 (error)

System action: The utility terminates.

User response: Use the -START DATABASE command to correct the logical error range (LPL) for the table space. Then resubmit the original failing utility.

DSNU206I *csect-name* **- LOGICAL ERROR RANGE (LOGICAL PAGE LIST) ON INDEX** *creator.index-name* **PROHIBITS PROCESSING**

Explanation: The utility cannot continue because it encountered a logical error range (logical page list) on the identified index space or partition.

Severity: 8 (error)

System action: The utility terminates.

User response: Issue the -DISPLAY command for the index to identify the logical error range (LPL). There are two ways to restore the index or index partition that has the LPL before resubmitting the original failed utility.

- Issue the START DATABASE command to clear the LPL status, then run the RECOVER or REBUILD INDEX utility.
- Run the RECOVER or REBUILD INDEX utility to restore the entire nonpartitioning index (NPI).

DSNU207I *csect-name* **- GROUP BUFFER POOL RECOVERY PENDING ON TABLE SPACE** *database.tablespace-name* **PROHIBITS PROCESSING**

Explanation: You cannot execute a utility on a table space that is in a group buffer pool recovery pending status.

Severity: 8 (error)

System action: Utility processing does not start.

User response: Use the -START DATABASE command to remove the table space from the group buffer pool recovery pending status. Then resubmit the job.

DSNU208I *csect-name* **- GROUP BUFFER POOL RECOVERY PENDING ON INDEX** *creator.index-name* **PROHIBITS PROCESSING**

Explanation: You cannot execute a utility on an index that is in a group buffer pool recovery pending (GRECP) status.

Severity: 8 (error)

System action: Utility processing does not start.

User response: Run the REBUILD INDEX utility on the index to remove the table space from the group buffer pool recovery pending status. Then resubmit the job.

DSNU209I *csect-name* **- RESTART PENDING ON** *obj-type database.tablespace-name* **PROHIBITS PROCESSING**

Explanation: An attempt was made to execute a utility against a table space or index space that is in a restart pending state.

Severity: 8 (error)

System action: Utility processing is not initiated.

System programmer response: Use the DISPLAY THREAD TYPE(POSTPONED) to determine the status of the resource. Use the RECOVER POSTPONED command if recover has not already been started. Resubmit the job.

DSNU210I *csect-name* **- INFORMATIONAL COPY PENDING ON** *obj-type obj-qual.obj-name* **PROHIBITS PROCESSING**

Explanation: An attempt was made to execute the RECOVER utility with the ERROR RANGE or PAGE options against an index in the informational copy pending state.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Use the REBUILD INDEX utility instead.

DSNU211I *csect-name* **CHECK PENDING ON** *obj-type obj-qual.obj-name* **PROHIBITS PROCESSING**

Explanation: The requested utility can not be run against an index in the check pending state. The index may be out of sync with its table space.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Run either the CHECK INDEX utility or the REBUILD INDEX utility. The CHECK INDEX utility will identify any errors (for example, if the index keys do not match the data in the table space.) If there are no errors, the check pending state will be reset. The REBUILD INDEX utility will rebuild the index from the data in the table space.

DSNU212I *csect-name* **- REORG PENDING ON TABLE SPACE** *database.index-name* **PROHIBITS PROCESSING**

Explanation: An attempt was made to execute a utility against a table space that is in a REORG PENDING state.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Execute REORG with SHRLEVEL NONE against the partitions in REORG PENDING state to rebalance the partitions and remove the REORG PENDING state.

DSNU213I *csect-name* **- AUX CHECK PENDING ON TABLE SPACE** *table-space* **PROHIBITS PROCESSING**

Explanation: The *processing* is prohibited because the *table-space* is in an AUX CHECK PENDING state.

System action: Utility terminates and returns completion code=8.

Programmer response:

Run the CHECK DATA utility with the AUXERROR

INVALIDATE option. If there are no AUX CHECK errors, AUX CHECK PENDING will be reset. If AUX CHECK errors are found, AUX CHECK PENDING will be reset and the table space will be set to an AUX WARNING state. This allows the rows with AUX CHECK errors to be corrected. Correct the errors and rerun CHECK DATA. If no more AUX CHECK errors are found, the AUX WARNING state will be reset.

DSNU214I *csect-name* **- REBUILD PENDING ON INDEX** *creator.index-name* **PROHIBITS PROCESSING**

Explanation: An attempt was made to execute a utility against an index that is in rebuild pending state.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Remove the index from the rebuild pending state by using REBUILD INDEX against the index, then resubmit the job.

DSNU215I *csect-name* **REFRESH PENDING ON** *ob-type database.objectname* **PROHIBITS PROCESSING**

Explanation: DB2 attempted to execute a utility against a table space or index that has refresh pending status.

Severity: 8 (error)

System action: Utility procesing is not initiated.

User response: If the object in refresh pending status is a table space, use the RECOVER utility to recover the table space to a prior point in time. You can also reload the table space with LOAD REPLACE. If the object in refresh pending status is an index, use the REBUILD INDEX utility to rebuild the index from the data.

DSNU231I *csect-name* **DICTIONARY WITH** *n* **ENTRIES HAS BEEN SUCCESSFULLY BUILT FROM** *m* **ROWS FOR TABLESPACE** *tablespace-name*

Explanation: A compression dictionary was built for the specified tablespace. This message is issued during the RELOAD phase of the LOAD utility and at the end of the UNLOAD phase of the REORG utility.

The variable *n* is the number of dictionary entries, and can be any of the following values: 512, 1024, 2048, 4096, or 8192. *n* depends on the number of rows being loaded or reorganized.

m is the number of rows used to build the dictionary.

Severity: 0 (informational)

System action: Processing continues.

DSNU232I *csect-name* **KEEPDICTIONARY REQUESTED BUT COMPRESS ATTRIBUTE NOT DEFINED FOR TABLE SPACE** *table-space-name*

Explanation: The KEEPDICTIONARY option was specified in the LOAD or REORG TABLESPACE utility statement. This keyword should only be used with table spaces that have the COMPRESS YES attribute. Otherwise, it is ignored. The COMPRESS column of SYSIBM.SYSTABLEPART contains the current compression status.

Severity: 4 (warning)

System action: Processing continues. If the dictionary exists, it is erased if LOAD REPLACE or REORG is specified.

DSNU233I *csect-name* **COMPRESSION DICTIONARY DOES NOT EXIST FOR TABLE SPACE** *table-space-name*

Explanation: The KEEPDICTIONARY option was specified in the LOAD or REORG TABLESPACE utility statement for a table space that does not have a compression dictionary.

Severity: 4 (warning)

System action: Processing continues. A dictionary is built for the table space.

DSNU234I *csect-name* **COMPRESSION REPORT FOR TABLE SPACE** *table-space-name*

noncmpbytes
: KB WITHOUT COMPRESSION

cmpbytes
: KB WITH COMPRESSION

percsave
: PERCENT OF THE BYTES SAVED FROM COMPRESSED DATA ROWS

pctrowcomp
: PERCENT OF THE LOADED ROWS WERE COMPRESSED

noncmppages
: PAGES REQUIRED WITHOUT COMPRESSION

cmppages
: PAGES REQUIRED WITH COMPRESSION

auncmprowlen
: BYTES FOR AVERAGE UNCOMPRESSED ROW LENGTH

avgcmprowlen
: BYTES FOR AVERAGE COMPRESSED ROW LENGTH

pagesave
: PERCENT OF THE DB2 DATA PAGES SAVED USING COMPRESSED DATA

Explanation: The data in the table space was compressed. This message is issued at the end of the RELOAD phase for the LOAD and REORG utility. The

statistics that follow apply only to the data that was loaded or reorganized. For example, RESUME(YES) reports statistics only on the additional rows that were loaded, not what was previously in the table space.

The compression statistics follow:

noncmpbytes
The total number of uncompressed KB (where KB = 1024 bytes) for the loaded or reorganized data rows (including the row prefix). This number was rounded to the nearest KB.

cmpbytes
The total number of compressed KB for the loaded or reorganized data rows (including the row prefix). This number was rounded to the nearest KB.

percsave
The percentage of bytes saved from the loaded or reorganized data rows as a result of compression.

The values above reflect only the number of bytes for the data row and the respective row prefix (6 bytes). DB2 page and dictionary page overhead is not included.

pctrowcomp
The percentage of the rows that were compressed within the total number of rows that were loaded or reorganized. The stored length and the external length of a row was compared to determine if the row was compressed. If an EDITPROC is involved, its effect on the length of the row is considered in the calculation.

noncmppages
The estimated number of DB2 pages required by the uncompressed data rows that were loaded or reorganized. Space required for PCTFREE and FREEPAGE is included. This number was rounded up to the next integer.

cmppages
The estimated number of DB2 pages required by the compressed data rows that were loaded or reorganized. Space required for PCTFREE and FREEPAGE is included. Dictionary pages are also included unless a LOAD RESUME(YES) was run. This number was rounded up to the next integer.

auncmprowlen
Average uncompressed row length.

avgcmprowlen
Average compressed row length.

pagesave
An estimate of the percentage of DB2 pages saved by compressing the loaded or reorganized data rows.

Severity: 0 (informational)

System action: Processing continues.

DSNU235I *csect-name* **THE COMPRESSION DICTIONARY WAS NOT BUILT FOR TABLE SPACE** *table-space-name*

Explanation: There is insufficient data to build a compression dictionary. For the LOAD utility, the input data set is empty or all of the input records were discarded. For the REORG TABLESPACE utility, the table space is empty. The data was not compressed.

Severity: 0 (informational)

System action: Processing continues.

User response: The input data set must contain data in order to build a compression dictionary while running the LOAD utility. Ensure that the input records are not discarded by the WHEN clause or conversion errors. For the REORG TABLESPACE utility, the table space must be populated with data in order to build a compression dictionary.

DSNU236I *csect-name* **DICTIONARY FOR TABLE SPACE** *table-space-name* **HAS BEEN ERASED.**

Explanation: The compression dictionary of the specified table space was erased.

The COMPRESS attribute of the table space is NO, and there are no active rows in this table space. In this case, the LOAD RESUME NO utility erases the dictionary.

Severity: 0 (informational)

System action: Processing continues.

| **DSNU237I** **TABLE** *table2* **IS NOT IN SAME TABLESPACE AS TABLE** *table1*

| **Explanation:** All source tables for a single LOAD utility statement must be in the same table space.

| **Severity:** 8 (error)

| **System action:** Utility processing terminates.

| **User response:** Remove the INTO TABLE control statement that specifies *table2* from the current LOAD request. Create another LOAD utility statement that contains the INTO TABLE control statement for *table2* so that all source tables in a single LOAD request are from the same table space. Resubmit the job.

DSNU241I *csect-name* **DICTIONARY WITH** *n* **ENTRIES HAS BEEN SUCCESSFULLY BUILT FROM** *m* **ROWS FOR TABLE SPACE** *table-space-name*, **PARTITION** *part-num*

Explanation: A compression dictionary was built for

the specified partition of the table space. This message is issued during the RELOAD phase of the LOAD utility and at the end of the UNLOAD phase of the REORG utility.

The variable *n* is the number of dictionary entries: 512, 1024, 2048, 4096, or 8192. *n* depends on the number of rows being loaded or reorganized.

m is the number of rows used to build the dictionary.

Severity: 0 (informational)

System action: Processing continues.

DSNU242I *csect-name* **KEEPDICTIONARY REQUESTED BUT COMPRESS ATTRIBUTE NOT DEFINED ON TABLE SPACE** *table-space-name*, **PARTITION** *part-num*

Explanation: The KEEPDICTIONARY option was specified in the LOAD or REORG TABLESPACE utility statement. This keyword should only be used with table space partitions that have the COMPRESS YES attribute. Otherwise, it is ignored. The COMPRESS column of SYSIBM.SYSTABLEPART contains the current compression status.

Severity: 4 (warning)

System action: Processing continues. If the dictionary exists, it is erased if LOAD REPLACE or REORG is specified.

DSNU243I *csect-name* **COMPRESSION DICTIONARY DOES NOT EXIST FOR TABLE SPACE** *table-space-name*, **PARTITION** *part-num*

Explanation: The KEEPDICTIONARY option was specified in the LOAD or REORG TABLESPACE utility statement for a table space partition that does not have a compression dictionary.

Severity: 4 (warning)

System action: Processing continues. A dictionary is built for the table space partition.

DSNU244I *csect-name* **COMPRESSION REPORT FOR TABLE SPACE** *table-space-name*, **PARTITION** *part-num*

noncmpbytes
KB WITHOUT COMPRESSION

cmpbytes
KB WITH COMPRESSION

percsave
PERCENT OF THE BYTES SAVED FROM COMPRESSED DATA ROWS

pctrowcomp
PERCENT OF THE LOADED ROWS WERE COMPRESSED

noncmppages
PAGES REQUIRED WITHOUT COMPRESSION

cmppages
PAGES REQUIRED WITH COMPRESSION

auncmprowlen
BYTES FOR AVERAGE UNCOMPRESSED ROW LENGTH

avgcmprowlen
BYTES FOR AVERAGE COMPRESSED ROW LENGTH

pagesave
PERCENT OF THE DB2 DATA PAGES SAVED USING COMPRESSED DATA

Explanation: The data in the table space partition was compressed. This message is issued at the end of the RELOAD phase for the LOAD and REORG utility. The statistics that follow apply only to the data that was loaded or reorganized. For example, RESUME(YES) reports statistics only on the additional rows that were loaded, not what was previously in the table space.

The compression statistics follow:

noncmpbytes
The total number of uncompressed KB (where KB = 1024 bytes) for the loaded or reorganized data rows (including the row prefix). This number was rounded to the nearest KB.

cmpbytes
The total number of compressed KB for the loaded or reorganized data rows (including the row prefix). This number was rounded to the nearest KB.

percsave
The percentage of bytes saved from the loaded or reorganized data rows as a result of compression.

The values above reflect only the number of bytes for the data row and the respective row prefix (6 bytes). DB2 page and dictionary page overhead is not included.

pctrowcomp
The percentage of the rows that were compressed within the total number of rows that were loaded or reorganized. The stored length and the external length of a row was compared to determine if the row was compressed. If an EDITPROC is involved, its effect on the length of the row is considered in the calculation.

noncmppages
The estimated number of DB2 pages required by the uncompressed data rows that were loaded or reorganized. Space required for PCTFREE and FREEPAGE is included. This number was rounded up to the next integer.

cmppages
The estimated number of DB2 pages required by the compressed data rows that were loaded or reorganized. Space required for PCTFREE and FREEPAGE is included. Dictionary pages are also included unless a LOAD RESUME(YES) was run. This number was rounded up to the next integer.

auncmprowlen
Average uncompressed row length.

avgcmprowlen
Average compressed row length.

pagesave
An estimate of the percentage of DB2 pages saved by compressing the loaded or reorganized data rows.

Severity: 0 (informational)

System action: Processing continues.

DSNU245I *csect-name* **THE COMPRESSION DICTIONARY WAS NOT BUILT FOR TABLE SPACE** *table-space-name*, **PARTITION** *part-num*

Explanation: There is insufficient data to build a compression dictionary. For the LOAD utility, the input data set is empty or all of the input records were discarded. For the REORG TABLESPACE utility, the partition is empty. The data was not compressed.

Severity: 0 (informational)

System action: Processing continues.

User response: The input data set must contain data in order to build a compression dictionary while running the LOAD utility. Ensure that the input records are not all being discarded by the WHEN clause or conversion errors. For the REORG TABLESPACE utility, the partition must be populated with data in order to build a compression dictionary.

DSNU246I *csect-name* **DICTIONARY FOR TABLE SPACE** *table-space-name*, **PARTITION** *part-num* **HAS BEEN ERASED.**

Explanation: The compression dictionary of the specified partition of the table space was erased.

The COMPRESS attribute of the table space partition is NO, and there are no active rows in this partition. In this case, the LOAD RESUME NO utility erases the dictionary.

Severity: 0 (informational)

System action: Processing continues.

DSNU250I *csect-name* **UNLOAD PHASE COMPLETE, ELAPSED TIME =** *hh/mm/ss*

Explanation: The UNLOAD phase of the REORG utility completed normally. The time that elapsed between the beginning of the REORG utility and the completion of the UNLOAD phase was hh/mm/ss, hours, minutes, and seconds, respectively.

Severity: 0 (informational)

System action: Action subsequent to this message depends on the UNLOAD option of the REORG utility statement:

- If the UNLOAD option is CONTINUE, execution of the utility continues into the RELOAD phase.
- If the UNLOAD option is PAUSE, execution of the utility is stopped at the beginning of the RELOAD phase.
- If the UNLOAD option is ONLY or EXTERNAL, the REORG utility is finished. Processing continues to the next utility in the utility job step, or, if there are no more utility statements, the utility job step terminates normally.

User response: The user's response depends on the UNLOAD option in the REORG utility statement.

- If the UNLOAD option is CONTINUE, no user response is required. The message is informational only.
- If the UNLOAD option is PAUSE, the REORG utility is stopped in the reload phase. To complete the REORG process, the user must explicitly restart the utility job step using the RESTART(PHASE) option.
- If the UNLOAD option is ONLY or EXTERNAL, the REORG utility has completed, and the requested output is on the data set identified by the DD statement of the UNLDDN REORG utility statement option.

Operator response: The operator's response depends on the UNLOAD option of the REORG statement:

- If the UNLOAD option is CONTINUE, no operator response is required. The execution of the REORG utility is proceeding.
- If the UNLOAD option is PAUSE, further processing is dependent upon restarting the utility job step with the RESTART(PHASE) option. Until that time, resources are held exclusively by the REORG utility job step.
- If the UNLOAD option is ONLY or EXTERNAL, the REORG utility processing is complete. The data set defined by the DD statement nominated by the UNLDDN option is available for installation disposition.

DSNU251I *csect-name* **UNLOAD PHASE STATISTICS. NUMBER OF RECORDS UNLOADED=** *mmmmmm* **FOR** *obj-type obj-qual.obj-name* **PART** *nn*

Explanation: This message is issued at the completion of the UNLOAD phase of the REORG utility. It gives the number of records that were unloaded *mmmmmm* for partition *nn* of the tablespace, index, or indexspace that was specified. The count can indicate the amount of processing.

Severity: 0 (informational)

System action: Only valid table records are unloaded. Dropped table record counts are not reported.

User response: The record count represents the current cardinality of the table or index that was indicated in the error message. Consider invoking the RUNSTATS utility and then rebinding the involved PLANs if this value differs greatly from the SYSTABLE.CARD value so that the SQL optimizer can bring the PLANs up to date.

Operator response: Count can be indicative of the amount of processing that remains.

DSNU252I *csect-name* **UNLOAD PHASE STATISTICS. NUMBER OF RECORDS UNLOADED=** *mmmmmm* **FOR** *obj-type obj-qual.obj-name*

Explanation: This message is issued at the completion of the UNLOAD phase of the REORG utility. It gives the number of records that were unloaded *mmmmmm* for the tablespace, index, or indexspace that was indicated in the error message.

Severity: 0 (informational)

System action: Only valid table or index records are unloaded. Dropped table record counts are not reported.

User response: Record count represents the current cardinality of the table or index that was specified in the error message. Invocation of the RUNSTATS utility and a subsequent rebinding of the involved PLANs should be considered if this value differs greatly from the SYSTABLE.CARD value, so that the SQL optimizer can bring the PLANs up to date.

Operator response: Count can be indicative of the amount of processing remaining.

DSNU253I *csect-name* **UNLOAD PHASE STATISTICS. NUMBER OF RECORDS** *action***=** *mmmmmm* **FOR TABLE** *table-name*

Explanation: This message is issued at the completion of the UNLOAD phase of the REORG utility. If *action* is "UNLOADED", it gives the number of records *mmmmmm* that were unloaded for the specified table name. If *action* is "DISCARDED", it gives the number of records *mmmmmm* that were discarded for the specified table name.

Severity: 0 (informational)

System action: Table records were unloaded or discarded.

DSNU254I *csect-name* **- RECORD COUNTS INCONSISTENT FOR** *pppppppp* **PHASE, NUMBER OF RECORDS EXPECTED =** *nnnnnnnn*, **NUMBER OF RECORDS PROCESSED =** *aaaaaaaa*

Explanation: This message is issued during the *pppppppp* phase for the REORG utility to report the condition where the number of records, *nnnnnnnn*, passed to the current phase is not equal to the number of records, *aaaaaaaa*, processed in this phase.

Severity: 4 (warning)

System action: Processing continues if UNLOAD PAUSE option was used.

User response: Check if unload data set was altered by user. Determine if the inconsistent record counts between the unload and reload phases is a potential problem. If it is an internal DB2 problem, report to system programmer.

System programmer response: Determine the failing environment. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: If the unload data set was manipulated by user during restart process, make sure it was handled properly. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 5.

DSNU255I *csect-name* **- KEY COUNT INCONSISTENT FOR** *pppppppp* **PHASE, NUMBER OF KEYS EXPECTED =** *nnnnnnnn*, **NUMBER OF KEYS PROCESSED =** *aaaaaaaa*.

Explanation: This message is issued during the *pppppppp* phase for the LOAD or REORG utility to report the condition where the number of keys, *nnnnnnnn*, passed to the current phase is not equal to the number of keys, *aaaaaaaa*, processed in this phase. One or more of the affected indexes may be inconsistent.

Severity: 8 (error)

System action: Processing stops and all indexes are placed in rebuild pending state.

User response: TERM the utility and re-access the

indexes by running REPAIR utility to reset the rebuild-pending states on indexes. Run the CHECK INDEX utility to determine if any of the affected indexes are inconsistent. Inconsistent indexes can be rebuilt by the REBUILD INDEX utility.

System programmer response: Determine the failing environment. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: If the working data set was manipulated by the user during the restart process, make sure it was handled properly. Refer to *DB2 Utility Guide and Reference*.

If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNU256I *csect-name* **INVALID PARTITION SPECIFICATION**

Explanation: This message is issued by the utility in response to an invalid PART option specification.

The PART option is invalid in the following situations:

- when it is specified for a nonpartitioned table space
- if the indicated PART integer is not pertinent to the partitions existing for the indicated table space
- or if a ROWID column is part of the partitioning key
- if LOAD INTO TABLE PART *x* is performed on a table space where an identity column is part of the partitioning key

The utility job step is terminated.

System action: The error is sufficient to terminate further processing of the utility job step. The utility job step is terminated, and resources are released to their previous state.

User response: Determine the correct partition specification for the table space that was the subject of the specified control utility statement, and resubmit the utility job step. Examine the SYSIBM.SYSTABLESPACE in the DB2 catalog for the correct range of partition numbers for the table space. If a ROWID column is part of the partitioning key, the PART option is invalid.

DSNU257I *csect-name* **UNABLE TO RESTART BUILD PHASE**

Explanation: Restarting in the BUILD PHASE or the SORTBLD phase of the LOAD utility with the RESUME(YES) option is not permitted.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Recover all indexes within the table space which were left in a rebuild pending state when the LOAD utility failed.

Problem determination: Refer to *DB2 Utility Guide and Reference* and Part 2 (Volume 1) of *DB2 Administration Guide* for a discussion of LOAD restarting constraints.

DSNU258I *csect-name* **BUILD PHASE STATISTICS. NUMBER OF INDEXES =** *mmm*

Explanation: This message is issued at normal completion of the BUILD phase of the LOAD or REORG or REBUILD INDEX utility. The number of indexes that were updated is indicated as 'mmm'.

Severity: 0 (informational)

System action: The BUILD phase of either the LOAD or REORG or REBUILD INDEX utility is completed. Upon completion of the utility job step, they are available for REFERENCE processing.

DSNU259I *csect-name* **BUILD PHASE COMPLETE, ELAPSED TIME =** *hh/mm/ss*

Explanation: This message is issued at the normal completion of the BUILD phase of the REORG, LOAD, and REBUILD INDEX utilities. The time that has elapsed between the beginning of the phase and the completion of this phase was 'hh' hours, 'mm' minutes, and 'ss' seconds.

System action: Processing continues to the next phase of the LOAD or REORG or REBUILD INDEX utility. If this is the last phase, processing continues to the next utility statement in the utility job step, if any.

DSNU260I *csect-name obj-type obj-qual.obj-name* **IS EMPTY**

Explanation: This message is issued as a warning when REBUILD INDEX(ALL) or REORG TABLESPACE are executed on an empty table space.

This message is also issued when COPY detects that the first space map page in the table space or partition is missing. This is an error condition that might be caused by a delete and redefine of the data set outside of DB2, or by a previous DB2 failure that occurred on the data set.

Severity: 4 (warning) or 8 (error)

System action: For REBUILD INDEX or REORG TABLESPACE, the return code of 4 will allow the next utility function in the job step, if any, to be initiated.

For COPY, the table space, index space, or partition being copied will be placed in recover pending status. The return code of 8 will prohibit further utility processing in the job step.

User response: Notify the system programmer if the return code is 8. If the return code is 4, no action is required.

System programmer response: If the return code is 8, the table space, indexspace, or partition being copied

should be recovered with the RECOVER utility.

DSNU269I *csect-name* **FIELD** *field-name* **IS NOT ALLOWED**

Explanation: You specified a ROWID GENERATED ALWAYS in the field specification list. Only a ROWID GENERATED BY DEFAULT is allowed to be specified in the field specification list.

Identity columns that were defined as GENERATED ALWAYS can not be specified in a LOAD field specification list, nor can these columns be implied in LOAD FORMAT UNLOAD or LOAD with no field specification list.

Severity: 8 (error)

System action: The LOAD utility is terminated

User response: Omit the ROWID GENERATED ALWAYS column from the LOAD field specification list and rerun the job.

DSNU270I *csect-name - table-name, column-name* **IN VIOLATION OF INSTALLATION DEFINED FIELD PROCEDURE** *procedure-name. function-code, return-code, reason-code, message-token*

Explanation: An installation-supplied field procedure has returned an error during encoding for LOAD or decoding for REORG. See SQLCODE -681 for more detail.

Severity: 8 (error)

System action: The utility will issue a system '04E' abend with a reason code of '00E40336'.

User response: Request the dump produced as a result of the error to the system programmer so that the cause of the error can be determined and corrected.

System programmer response: Examine the field procedure to determine why an error return code was generated. Refer to *DB2 Utility Guide and Reference* for more information.

DSNU271I *csect-name* **- REORG PENDING ON FOR TABLE SPACE** *database.index-name* **PART** *part number* **PROHIBITS PROCESSING**

Explanation: An attempt was made to execute a REORG utility to redistribute data in a partitioned table space. The partition number stated in this message was found to have the REORP state on, but was not specified on the PART *n1*:*n2* parameter of the REORG utility.

If you are running REORG to materialize identity column values, you must reorganize the entire table space.

Severity: 8 (error)

System action: Utility processing is not initiated.

System programmer response: Identify the correct partition range specification for the REORG utility by issuing the DISPLAY command for the table space, and resubmit the utility job step.

DSNU272I *csect-name* **- FIELD PROCEDURE** *procedure-name* **COULD NOT BE LOADED**

Explanation: An installation supplied field procedure could not be loaded during encoding for LOAD or decoding for REORG. See SQLCODE -682 for more detail.

Severity: 8 (error)

System action: The utility will terminate with a return code of 8.

User response: Take the job output to the system programmer.

System programmer response: Determine why the field procedure could not be loaded.

DSNU273I *csect-name* **- REORG WITH SHRLEVEL REFERENCE OR CHANGE CANNOT APPLY TO TABLE SPACE** *name* **PART** *n1:n2*

Explanation: REORG with SHRLEVEL REFERENCE or CHANGE cannot be used on table space partitions with REORG PENDING restrictive state.

Severity: 8 (error)

System action: The REORG utility job is terminated.

User response: Execute REORG with SHRLEVEL NONE against the partitions in REORG PENDING state to rebalance the partitions and remove the REORG PENDING state.

DSNU280I *csect-name* **- ERROR=X'reason-code' FROM EDIT ROUTINE** *procedure-name* **FOR** *table-name*

Explanation: The edit routine provided by your installation has returned an error during the load of a record into the table indicated. The reason code is set by the edit routine.

Severity: 8 (error)

System action: The LOAD or REORG utility will receive ABEND04E RC00E40323.

User response: Correct any errors in the input data. Ensure that the edit routine provided is correct. The utility can be restarted or terminated.

System programmer response: Verify that the edit routine is correct.

DSNU281I *csect-name* **RECORD** *(n)* **OF TABLE** *table-name* **VIOLATING VALID ROUTINE** *validproc-name*, **ERROR CODE=X'reason-code'**

Explanation: The validation exit routine provided by your installation blocked the load of record 'n' into TABLE 'table-name'. The reason-code was set by the exit routine.

Severity: 4 (warning)

System action: The LOAD utility continues to process the following input record.

User response: Check the error input records against the error reason code returned from the validation procedure. Refer to *DB2 Utility Guide and Reference* for a definition of LOAD input field column specifications.

System programmer response: Verify that the validation routine is correct.

DSNU282I *csect-name* **UNABLE TO RESTART AT LAST CHECKPOINT WITH DDNAME** *'dddddddd'*

Explanation: The data set with DDNAME 'dddddddd' is a concatenated data set and the utility can be restarted only at the beginning of the phase (RESTART PHASE), not at the last checkpoint (RESTART CURRENT).

Severity: 8

System action: The utility will issue a system '04E' abend with a reason code of '00E40323'.

User response: Change the RESTART option to RESTART PHASE and resubmit the job step.

Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 5.

DSNU283I *csect-name type-of-error* **SQLCODE =** *sqlcode*, **SQLERRM =** *sqlerrm*, **SQLSTATE =** *sqlstate*, **SQLERRP =** *sqlerrp*, **SQLERRD =** *sqlerrd1 sqlerrd2 sqlerrd3 sqlerrd4 sqlerrd5 sqlerrd6*

Explanation: An error occurred, that returned SQLCA error information to the utility. This message contains multiple components. The SQL diagnostic information can be found in the SQLERRD1, SQLERRD2, SQLERRD3, SQLERRD4, SQLERRD5, and SQLERRD6 fields.

type-of-error
Indicates a trigger error or a LOB error.

sqlcode The SQL return code. If the SQL code is negative, then this indicates that an error has occurred. For more information about this SQL return code, see DB2 Codes.

sqlerrp The name of the procedure that detected the error.

sqlerrm A length indicator and one or more tokens, separated by X'FF', that are substituted for variables in the descriptions of error conditions. It may contain truncated tokens. A length of 70 bytes might indicate a possible truncation.

sqlstate A return code for the outcome of the most recent execution of an SQL statement.

sqlerrd Each field contains a value that indicates the number of rows that qualified for the SQL statement, the number of rows in the result table, a rough estimate of required resources, or an internal error code.

For a complete description of sqlerrp, sqlerrm, and the sqlerrd fields, see the description of SQLCA fields in .

Severity: 8 (error)

System action: Processing terminates.

User response: Determine the cause for the SQL error by using information about the specific SQLCODE. Correct the error, and rerun the utility.

| **DSNU284I** *csect-name*- **LOAD UTILITY NOT ALLOWED AGAINST A SYSTEM-MAINTAINED MATERIALIZED QUERY TABLE** *table-name*

| **Explanation:** An attempt has been made to run the LOAD utility against a system-maintained materialized query table. This is not allowed.

| **Severity:** 8

| **System action:** Utility processing terminates.

| **User response:** If the table name was incorrect, correct the table name and resubmit the utility job step. Otherwise, take one of the following actions:

| • Use the REFRESH TABLE SQL statement to insert data into the system-maintained materialized query table.

| • Use the ALTER statement to change the materialized query table to a user-maintained materialized query table or a base table, and resubmit the utility job step.

DSNU286I *csect-name* **REORG TABLESPACE** *dbname.tsname* **OFFPOSLIMIT SYSINDEXPART ROWS**

Explanation: This is the header message for the SYSINDEXPART section of the REORG TABLESPACE OFFPOSLIMIT output. This section lists the SYSINDEXPART rows for the table space or table space partition as specified in the REORG TABLESPACE

input statement. The fields from SYSINDEXPART are:

CREATOR
Index creator name

IXNAME
Index name

CREATOR
Table creator name

TBNAME
Table name

PART Partition number

CARDF
Number of rows referenced (floating point)

FAROFFPOSF
Number of rows referenced afar (floating point)

NEAROFFPOSF
Number of rows referenced nearby (floating point)

STATSTIME
TIMESTAMP of RUNSTATS update

Rows which meet the OFFPOSLIMIT criteria are prefaced with '*'.

Severity: 0 (informational)

System action: Processing continues normally

User response: If the TIMESTAMP reported in the STATSTIME column is not recent, the OFFPOSLIMIT criteria might be evaluated with obsolete statistics.

DSNU287I *csect-name* **REORG TABLESPACE** *dbname.tsname* **INDREFLIMIT SYSTABLEPART ROWS**

Explanation: This is the header message for the SYSTABLEPART section of the REORG TABLESPACE INDREFLIMIT output. This section lists the SYSTABLEPART rows for the table space or table space partition as specified in the REORG TABLESPACE input statement. The fields from SYSTABLEPART are:

DBNAME
Database name

TSNAME
Table space name

PART Partition number

CARD Number of rows in table space

FARINDREF
Number of rows relocated afar

NEARINDREF
Number of rows relocated nearby

STATSTIME
TIMESTAMP of RUNSTATS update

Rows which meet the INDREFLIMIT criteria are prefaced with '*'.

Severity: 0 (informational)

System action: Processing continues normally

User response: If the TIMESTAMP reported in the STATSTIME column is not recent, the INDREFLIMIT criteria might be evaluated with obsolete statistics.

DSNU288I *csect-name* **REORG INDEX** *dbname.ixname* **LEAFDISTLIMIT SYSINDEXPART ROWS**

Explanation: This is the header message for the SYSINDEXPART section of the REORG INDEX LEAFDISTLIMIT output. This section lists the SYSINDEXPART rows for the index or index partition as specified in the REORG INDEX input statement. The fields from SYSINDEXPART are:

CREATOR
Index creator name

IXNAME
Index name

PART Partition number

LEAFDIST
Number of pages between leaf pages multiplied by 100

STATSTIME
TIMESTAMP of RUNSTATS update

Rows which meet the LEAFDISTLIMIT criteria are prefaced with '*'.

Severity: 0 (informational)

System action: Processing continues normally

User response: If the TIMESTAMP reported in the STATSTIME column is not recent, the LEAFDISTLIMIT criteria might be evaluated with obsolete statistics.

DSNU289I *csect-name type* **REORG LIMITS HAVE BEEN MET**

Explanation: The REORG utility is indicating whether any specified limits have been met. For REORG TABLESPACE, this might be the OFFPOSLIMIT or the INDREFLIMIT. For REORG INDEX, this is the LEAFDISTLIMIT. *type* values:

NO If no limits are met.

blank If at least one limit is met.

Severity: The return codes are:

1 (informational)
If no limit is met.

2 (informational)
If at least one limit is met.

System action: Utility processing continues.

User response: If the REPORTONLY option is specified and the result of this job step is not used to conditionally execute other job steps, examine the utility messages for this table space, partition, or index to determine if a REORG should be performed.

DSNU290I *csect-name type* **REORG WILL BE PERFORMED**

Explanation: The REORG utility is indicating whether any specified limits have been met and whether a REORG will be performed. For REORG TABLESPACE, this might be the OFFPOSLIMIT or the INDREFLIMIT. For REORG INDEX, this is the LEAFDISTLIMIT. *type* values:

NO If no limits are met.

blank If at least one limit is met.

Severity: 0 (informational)

System action: Utility processing continues. A REORG is performed if any limits are met.

DSNU291I **MAXIMUM COMPOSITE UNLOAD RECORD SIZE TOO LARGE FOR SORT. SORTDATA KEYWORD IS IGNORED.**

Explanation: The SORTDATA keyword was specified for a REORG utility job. However, the largest possible record in the table space, prefixed by the largest key, produces a composite record that is too large to be processed by the SORT component. The SORTDATA keyword is ignored.

This message is issued by the following CSECT: DSNURFIT

Severity: 0

System action: Processing continues normally.

User response: If you must use SORTDATA for the table space, try reducing the size of either the largest record or key field by redefining the table. In some cases, changing a field from LONG VARCHAR to VARCHAR with a specified maximum length might help.

Problem determination: This problem should only occur on a table space with a 32KB page size. Analyze the requirement for maximum data length in VARCHAR and LONG VARCHAR columns to see if a shorter length can be specified. It might be necessary to remove the SORTDATA keyword from the utility statement.

DSNU292I *csect-name* **- -ALTER UTILITY CANNOT APPLY TO THIS UTILITY, REASON=***c*

Explanation: The -ALTER UTILITY command only applies to a job that is executing REORG with SHRLEVEL REFERENCE or CHANGE. In addition, if -ALTER UTILITY specifies MAXRO, LONGLOG, or DELAY, the job must be executing REORG with SHRLEVEL CHANGE.

1 The job is not executing REORG with SHRLEVEL REFERENCE or CHANGE.

2 -ALTER UTILITY specifies MAXRO, LONGLOG, or DELAY, but the job is executing REORG with SHRLEVEL REFERENCE.

Severity: 8 (error)

System action: -ALTER UTILITY does not execute.

User response: Apply -ALTER UTILITY to a job that is executing the proper utility.

DSNU293I *csect-name* **- REORG WITH SHRLEVEL REFERENCE OR CHANGE CANNOT APPLY TO THE** *name* **TABLE SPACE**

Explanation: REORG with SHRLEVEL REFERENCE or CHANGE cannot operate on the following catalog and directory table spaces: DSNDB06.SYSDBASE, DSNDB06.SYSDBAUT, DSNDB06.SYSGROUP, DSNDB06.SYSPLAN, DSNDB06.SYSVIEWS, and DSNDB01.DBD01.

Severity: 8 (error)

System action: REORG does not execute.

User response: Execute REORG with SHRLEVEL NONE.

DSNU294I *csect-name* **- LARGEST COMPOSITE RECORD IS TOO BIG FOR REORG WITH SHRLEVEL CHANGE**

Explanation: SHRLEVEL CHANGE was specified for a REORG utility job, and the table has a clustering index. However, the largest possible record in the table space, prefixed by the largest key, produces a composite record that is too large to be processed by the SORT component. See the computation of the length under REORG in the *DB2 Utility Guide and Reference*. If the largest possible composite record to be sorted exceeds 32760 bytes in length, which is the maximum record size for a BSAM data set, REORG cannot operate if SHRLEVEL CHANGE is specified.

Severity: 8 (error)

System action: REORG does not execute.

User response: Execute REORG with SHRLEVEL NONE or REFERENCE and specify SORTDATA NO. If you must use SHRLEVEL CHANGE for the table space, try reducing the size of either the largest record or key

field by redefining the table. In some cases, changing a field from LONG VARCHAR to VARCHAR with a specified maximum length might help.

Problem determination: This problem should only occur on a table space with a 32KB page size. Analyze the requirement for maximum data length in VARCHAR and LONG VARCHAR columns to see whether a shorter length can be specified.

DSNU295I *csect-name* **- IN UTILTERM PHASE OF REORG WITH SHRLEVEL REFERENCE OR CHANGE, DATA SET** *name* **COULD NOT BE DELETED**

Explanation: For the area (table space or partition) being reorganized, REORG with SHRLEVEL REFERENCE or CHANGE unloads data from the original data sets and reloads data into shadow data sets. REORG then exchanges the names of the original and shadow data sets. The original name of a data set is in the form "*catname*.DSNDB*x*.*dbname*.*psname*.I0001.A*nnn*". The shadow name of a data set is in the form "*catname*.DSNDB*x*.*dbname*.*psname*.J0001.A*nnn*". REORG uses such data sets for indexes as well as data.

If a data set is DB2-managed, the utilterm phase of REORG deletes the shadow data set, which was the original data set. This message indicates that the shadow data set *name* could not be deleted.

Severity: 4 (warning)

System action: REORG continues execution, but it does not delete the specified data set.

User response: Try to delete the data set. If you cannot delete it, notify the system programmer.

DSNU296I *csect-name* **- IN REORG WITH SHRLEVEL REFERENCE OR CHANGE, DATA SET** *name* **DOES NOT EXIST, CONTEXT=***c*

Explanation: For the area (table space or partition) being reorganized, REORG with SHRLEVEL REFERENCE or CHANGE unloads data from the original data sets and reloads data into shadow data sets. REORG then exchanges the names of the original and shadow data sets, using temporary names during the renaming. The original name of a data set is in the form "*catname*.DSNDB*x*.*dbname*.*psname*.I0001.A*nnn*". The shadow name of a data set is in the form "*catname*.DSNDB*x*.*dbname*.*psname*.J0001.A*nnn*". The temporary name of a data set is in the form "*catname*.DSNDB*x*.*dbname*.*psname*.T0001.A*nnn*". REORG uses such data sets for indexes as well as data.

During its initialization, REORG cannot execute if:

- Data sets with the temporary names already exist
- DB2-managed data sets with the shadow names already exist
- User-managed data sets with the shadow names do not already exist

To exchange the names of the original and shadow data sets, REORG renames in the following ways:

1 In the switch phase, it renames the original data set from the original name to the temporary name.

2 In the switch phase, it renames the shadow data set from the shadow name to the original name.

3 In the utilterm phase, it renames the original data set from the temporary name to the shadow name (for user-managed data sets). It deletes the temporary name for DB2-managed data sets.

If either a -TERM UTILITY command is issued, or the REORG switch phase is restarted after REORG has begun renaming data sets, the names revert back to their original form prior to the utilterm phase of REORG:

The context code indicates the specific type of error:

1

During the switch phase of REORG, on rename 1, REORG detected that a data set with an original name does not exist. This error condition can arise for a DB2-managed or user-managed data set.

During the switch phase restart or a -TERM UTILITY during the switch phase, on rename 4, REORG detected that a data set with an original name does not exist. This error condition can arise for a DB2-managed or user-managed data set.

2

During the switch phase of REORG, on rename 2, REORG detected that a data set with a shadow name does not exist. This error condition can arise for a DB2-managed or user-managed data set.

During the switch phase restart or a -TERM UTILITY during the switch phase, on rename 5, REORG detected that a data set with a shadow name does not exist. This error condition can arise for a DB2-managed or user-managed data set.

3 During the utilterm phase of REORG, on rename 3, REORG detected that a data set with a temporary name does not exist. This error condition can arise for a user-managed data set.

7 During initialization of REORG, REORG detected that the shadow data set did not

exist. This error can only occur with a table space, index, or partition that is stored in user-managed data sets.

Severity: 4 (warning) or 8 (error)

System action: For context codes 1 or 2, utility processing is abended. ABEND reason code 00E4031B is issued. The object is left in a recovery pending state.

For context code 3, the utility job step that caused the function to be requested continues with no renaming. The utility job is placed in the stopped state. The object is left in a recovery pending state, and a message will be issued.

For context code 7, REORG does not execute.

For context codes 1, 2, or 3, the following messages are issued:

- If the data set is part of a nonpartitioned table space, the object is the table space, and the message is DSNU562I.
- If the data set is part of a partition of a table space, the object is the partition, and the message is DSNU560I.
- If the data set is part of a nonpartitioned index, the object is the index, and the message is DSNU557I.
- If the data set is part of a partition of a partitioned index, the object is the partition of the index, and the message is DSNU559I.

User response: For context codes 1 or 2, execute the RECOVER TABLESPACE or RECOVER or REBUILD utilities.

For context code 3, do nothing.

For context code 7, define all shadow objects prior to running REORG.

DSNU297I *csect-name* **COMPOSITE RECORD SIZE TOO LARGE FOR** *table-name*

Explanation: The size of the record that would be unloaded for REORG DISCARD or REORG UNLOAD EXTERNAL is greater than the largest allowed data set logical record length (32760 bytes).

Severity: 8 (error)

System action: The utility will terminate with a return code of 8.

User response: If you are attempting to discard records during REORG, you will have to use SQL to DELETE the records prior to the REORG. If you are attempting to unload records in an external format, you must use an application instead. If you are using the SORTDATA option with REORG UNLOAD EXTERNAL or REORG DISCARD, try removing the SORTDATA keyword.

Problem determination: This problem will most likely occur only on a table with BLOB, CLOB, or DBCLOB columns.

DSNU298I *csect-name* **- RESTART NOT ALLOWED IN** *phase-name* **PHASE OF REORG UTILITY WITH SHRLEVEL CHANGE**

Explanation: An attempt was made to restart the REORG utility with SHRLEVEL CHANGE in the *phase-name* phase. DB2 does not allow you to restart the REORG utility with SHRLEVEL CHANGE in that phase.

Severity: 8 (error)

System action: Utility processing abends. Abend reason code 00E40018 or 00E40310 might be issued.

User response: Issue the -TERM UTILITY command to terminate the utility job; then rerun the job without the RESTART parameter.

System programmer response: If necessary, take actions to prevent access to the table space or partition before terminating the utility.

DSNU299I *csect-name* **CHARACTER CONVERSION FROM CCSID** *from-ccsid* **TO CCSID** *to-ccsid* **FAILED WITH ERROR CODE 'error-code' FOR** *object-type object-name-qualifier.object-name column column-name***.**

Explanation: A conversion error occurred during the conversion of a string to a different coded character set. The error occurred during the processing of the utility statement.

The *from-ccsid* identifies the coded character set of the string to be converted.

The *to-ccsid* identifies the coded character set to which it must be converted.

The *error-code* indicates the type of error as defined as follows:

- 8 - Length exception (for example, expansion required for PC MIXED data exceeds the maximum length of the string).
- 12 - Invalid code point.
- 16 - Form exception (for example, invalid MIXED data).
- 20 - Translate procedure error (for example, an error during Unicode/390 stage 2 conversions or an exit set the length control field of the string to an invalid value).
- 24 - the conversion services have not been setup correctly or the conversion is not supported.

The *object_type* gives the type of the failed object. It could be a USERID or a SQLID or a TABLE with the conversion error. This could be either the target or the source of the conversion.

The *object-name-qualifier.object-name* is applicable for object type TABLE only. If the failed object type is TABLE, then the object qualifier name will be the table creator id and the object name will be the table name.

The *column* is applicable only if the failed object is TABLE. 'COLUMN' will be given immediately before the column-name is given.

The *column-name* gives the column name of the failing object. This could be either the target or the source of the conversion.

Severity: 8 (error)

System action: The utility job is terminated.

User response: Refer to *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem. The user may generate a dump by adding the DIAGNOSE ABEND statement in the utility job and re-running the utility. If the error can be fixed, correct the error and resubmit the utility, else issue the TERM utility command to terminate the utility.

DSNU300I *csect-name* **(RE)LOAD PHASE COMPLETE, ELAPSED TIME =** *hh/mm/ss*

Explanation: The reload phase of the REORG or LOAD utility has completed normally. The elapsed time for the execution was 'hh' hours, 'mm' minutes, and 'ss' seconds.

DSNU301I *csect-name* **- KEYWORD 'SORTDATA' SPECIFIED AND/OR KEYWORD 'NOSYSREC' WAS SPECIFIED BUT NO CLUSTERINGINDEX EXISTS, KEYWORD IS IGNORED**

Explanation: The 'SORTDATA' keyword was specified or the 'NOSYSREC' keyword was specified for a REORG utility job, or implied by the use of SHRLEVEL CHANGE. But it was found that no clustering index existed for the table space being reorganized. The keyword is ignored and processing continues normally.

Severity: 0

System action: Processing continues normally.

User response: If a clustering index was expected to be found for the table space being reorganized, determine the reason the index could not be found.

Problem determination: Probably a user error. Determine if a cluster index should be defined on one of the tables in the table space being reorganized. If so, define the cluster index; otherwise, remove the 'SORTDATA' and/or 'NOSYSREC' keyword from the utility statement if necessary.

DSNU302I *csect-name* **(RE)LOAD PHASE STATISTICS - NUMBER OF INPUT RECORDS PROCESSED =** *nnnnnnnn*

Explanation: This message is issued at the normal completion of the RELOAD phase of the REORG utility and the LOAD phase of the LOAD utility.

It indicates the number of records processed from the input record data set (that is, the data set identified either by the JCL DD statement identified in the UNLDDN option for REORG or the INDDN option for LOAD).

Severity: 0 (informational)

System action: The LOAD or REORG utility completes the phase in which the records are (re-)introduced into the DB2 database. This is also the last phase of the utility unless there are some indexes that need to be built and/or updated.

User response: The user can verify that the number of records (re-)loaded is consistent with expectations. For example, in REORG it matches the total number of records unloaded. For LOAD, it matches the number of records submitted minus the number of records reported rejected.

DSNU303I *csect-name* **(RE)LOAD PHASE STATISTICS - NUMBER OF RECORDS=** *nnnnnn* **FOR TABLE** *table-name* **PART=***part-number*

Explanation: This message is issued at the normal completion of the RELOAD phase of the REORG utility and the LOAD phase of the LOAD utility.

It indicates the number of records loaded into PART 'part-number' of 'table-name' from the source record data set. This data set is identified by the JCL DD statement in the UNLDDN option for REORG, or the INDDN option for LOAD.

This message is repeated for each INTO TABLE PART clause in the LOAD utility statement or each TABLESPACE PART clause in the REORG utility statement.

Severity: 0 (informational)

System action: The LOAD or RELOAD phase maintains a separate count for each table part. Rejected rows are not included in the count. Each count is reported as a separate message at the end of the LOAD or RELOAD phase.

The user can verify the number of records loaded or reloaded into the table part. In REORG, it should match the number of records unloaded for that table part. In LOAD, it should match the number of records processed.

DSNU304I *csect-name* **(RE)LOAD PHASE STATISTICS - NUMBER OF RECORDS =** *nnnnnn* **FOR TABLE** *table-name*

Explanation: This message is issued at the normal completion of the RELOAD phase of the REORG utility and the LOAD phase of the LOAD utility.

It indicates the number of records loaded into TABLE 'table-name' from the source record data set (that is, the data set identified by the JCL DD statement identified in the UNLDDN option for REORG, or the INDDN option for LOAD).

This message is repeated for each TABLE loaded by the current REORG or LOAD utility statement.

If the LOAD or REORG was restarted in a LOAD or RELOAD phase from the last commit point (i.e., the 'RESTART' parm was specified), 'nnnnnn' indicates the number of records loaded after the restart.

Severity: 0 (informational)

System action: The LOAD or RELOAD phase maintains a separate count for each table populated by this phase. Rejected rows are not included in the count. Each count is reported as a separate message at the end of the LOAD or RELOAD phase.

The user can verify that the number of records (re-)loaded into the table is consistent with expectations. For example, in REORG it matches the number of records unloaded for that table. For LOAD, it matches the number of records processed.

DSNU305I *csect-name* **- TABLE=***table-name* **HAS AN INCOMPLETE TABLE DEFINITION**

Explanation: Table 'table-name' does not have an index defined to enforce the uniqueness of its primary key or unique key.

Severity: 8 (error)

System action: Processing stops. No table in the table space is loaded.

User response: The user can resubmit the LOAD job after performing one of the following:

- Creating a unique index over the table's primary key
- Creating a unique index over the table's unique key(s)
- Dropping the definition of the table's primary key
- Dropping the definition of the table's unique key(s)

Problem determination: Issue SQL SELECT against the catalog to determine which indexes and relations exist against 'table-name'. Determine whether uniqueness of the primary key or unique key has been guaranteed through omission or through faulty definition of an index.

DSNU306I *csect-name* **- UNABLE TO DRAIN WRITERS FROM** *object-type object-name*

Explanation: The utility was unable to drain the writers from an object because the resource was unavailable. The object is identified by type and name. Message DSNT500I tells why the resource was unavailable. If the QUIESCE utility issues this message, DSNT500I is not issued, but message DSNT501I might be issued.

Severity: 8 (error)

System action: Utility processing terminates.

Problem determination: Refer to message DSNT500I or DSNT501I.

DSNU307I *csect-name* **PARENT TABLE** *parent-table-name* **OF** *table-name* **IN RELATIONSHIP** *relationship-name* **HAS AN INCOMPLETE TABLE DEFINITION**

Explanation: The 'table-name' identifies the target table of the job for the LOAD utility or the dependent table to be checked by the CHECK DATA utility. The 'parent-table-name' identifies the parent table of the target table for the LOAD utility or the dependent table for the CHECK DATA utility in RELATIONSHIP 'relationship-name'. The primary index of PARENT TABLE 'parent-table-name' has been dropped.

This message is repeated for each incompletely defined parent table in each relationship where the target table is a child.

Severity: 8 (error)

System action: The LOAD and CHECK DATA utilities stop further processing. No data is loaded by the LOAD utility and no relationships are checked by the CHECK DATA utility.

User response: Check the existence of the primary index and its parent tables identified by RELATIONSHIP 'relationship-name'.

Problem determination: If the incompletely defined tables were targets of the LOAD job, message DSNU305I has been issued for each such table. Refer to the catalog to determine if any tables in the table space that were not targets of the LOAD job are incompletely defined.

DSNU308I *csect-name* **INDEX KEY LIST RECORD LENGTH EXCEEDS MAXIMUM KEY BUFFER SIZE**

Explanation: The length of the index key list that will be built in the Data Base Services address space while loading a data record may be larger than the maximum key buffer size. The LOAD statement may be too complex.

Severity: 8 (error)

System action: Utility processing is terminated.

User response: Examine the preceding DSNU650I INTO TABLE statements. The key size and number of indexes affected by each INTO TABLE statement are accumulative. The combination of these elements may produce a list of index keys, including header bytes, that is too large to be contained within the maximum size buffer area during the load of each input data record. It may be necessary to reduce the number of INTO TABLE statements or to drop some of the indexes that are affected by the load.

DSNU309I *csect-name* **NOT ALL REQUIRED UNIQUE INDEXES HAVE BEEN DEFINED FOR TABLE** *table-name*

Explanation: The table 'table-name' was created with unique index requirements. Not all of these indexes have been defined.

Severity: 8

System action: Processing stops. No table in the table space is loaded.

User response: The user may resubmit the LOAD job after either:

- Creating all the required unique indexes, or
- Dropping the table and re-creating it without the unique index requirements.

Problem determination: Determine which unique indexes need to be created. Determine if the uniqueness requirement for the table is valid.

DSNU310I *csect-name* **RECORD 'n' WILL BE DISCARDED DUE TO** *field-name* **CONVERSION ERROR FOR** *table-name*

Explanation: The load utility found the input data provided for a field to be incorrect. The corresponding input data set record 'n' will be placed in the DISCARDS data set.

If the NOSUBS keyword was specified, it is possible that the record was discarded because the input contained a code point that when translated from the source CCSID to the target CCSID resulted in a substitution character being placed in the target string.

Severity: 4 (warning)

System action: The input data set record is noted and will be discarded in the DISCARD phase. The LOAD utility continues to process additional records in the input data set.

User response: Following the completion of the current LOAD utility invocation, examine the record in the DISCARDS data set. Correct the error in the discarded input data. Submit a RESUME(YES) LOAD invocation with the DISCARDS data set as the input data set. Refer to *DB2 Utility Guide and Reference* for a definition of LOAD input field column specifications.

DSNU311I *csect-name* **RECORD 'n' WILL BE DISCARDED DUE TO INVALID** *field-name* **COLUMN SPECIFICATION**

Explanation: The load utility calculated the location of an input data value using the field specifications supplied and VARCHAR length fields present within the input data set record. The calculated position for the field was determined to be incorrect.

Severity: 4 (warning)

System action: The input data set record number is noted and will be discarded in the DISCARD phase. The LOAD utility continues to process additional records in the input data set.

User response: Following the completion of the current LOAD utility invocation, examine the record in the DISCARDS data set. Correct the error in the discarded input data, if one exists. Otherwise, correct the input field specification. Submit a RESUME(YES) LOAD invocation, which uses the DISCARDS data set and the verified LOAD utility statement and field specifications. Refer to *DB2 Utility Guide and Reference* for a definition of LOAD input field column specifications.

DSNU312I *csect-name* **RECORD 'n' WILL BE DISCARDED DUE TO ERROR = X'reason-code' FROM EXIT ROUTINE** *procedure-name* **FOR** *table-name*

Explanation: The exit routine provided by your installation blocked the load of record 'n' into the table indicated. The reason-code was set by the exit routine.

Severity: 4 (warning)

System action: The input data set record number is noted and will be discarded in the DISCARD phase. error is placed in the DISCARDS data set. The LOAD utility continues to process additional records in the input data set.

User response: Following the completion of the current LOAD utility invocation, examine the record in the DISCARDS data set. Correct the error in the discarded input data, if one exists. Ensure that the exit routine provided is correct. Submit a RESUME(YES) LOAD invocation, which uses the DISCARDS data set and the verified LOAD utility statement and field specifications. Refer to *DB2 Utility Guide and Reference* for a definition of LOAD input field column specifications.

System programmer response: Verify that the validation routine is correct.

DSNU313I *csect-name* **RECORD 'n' WILL BE DISCARDED DUE TO PART OR HIGH KEY VALUESPECIFICATION(S)**

Explanation: The discarded record does not belong to any of the partitions being loaded. For a LARGE table space, this might be because the key is higher than the high key value specified for the last partition.

Severity: 4 (warning)

System action: The input data set record number is noted and will be discarded in the DISCARD phase. The LOAD utility continues to process additional records in the input data set.

User response: Following the completion of the current LOAD utility invocation, examine the record in the DISCARDS data set. Correct the error in the discarded input data, if one exists. Submit a RESUME(YES) LOAD invocation with the DISCARDS data set as the input data set. Refer to Part 2 of *DB2 Utility Guide and Reference* for a definition of LOAD input field column specifications.

System programmer response: Verify that the validation routine is correct.

DSNU314I *csect-name* **RECORD 'n' WILL BE DISCARDED DUE TO WHEN SPECIFICATION(S)**

Explanation: The discarded record satisfies none of the WHEN selection expressions provided.

Severity: 4 (warning)

System action: The input data set record is noted and will be discarded in the DISCARD phase. The LOAD utility continues to process additional records in the input data set.

User response: Following the completion of the current LOAD utility invocation, examine the record in the DISCARDS data set. Correct the error in the discarded input data, if one exists. Submit a RESUME(YES) LOAD invocation with the DISCARDS data set as the input data set. Refer to *DB2 Utility Guide and Reference* for a definition of LOAD input field column specifications. If FORMAT UNLOAD was specified on the load statement, the WHEN specifications are built internally. This specification ensures that the rows are loaded back into the exact same tables from which the data was unloaded. Refer to *DB2 Utility Guide and Reference* for a description of the FORMAT UNLOAD option of LOAD.

DSNU315I *csect-name* **RECORD 'n' WILL BE DISCARDED DUE TO UNEXPECTED ERROR**

Explanation: The LOAD of input data set record number 'n' was attempted but was not successful due to an unexpected error.

Severity: 8 (error)

System action: If the error was caused because a resource was not available, a DSNT500I message will be issued.

The utility step will be abended with a system X'04E' and an abend reason code of '00E40318'. If a DSNT500I message is not issued, an SVC dump will be requested. A SYSABEND dump of batch storage is also requested.

User response: If a DSNT500I message was issued, refer to the description under that message number. If the unavailable resource becomes available, the utility step is restartable.

If there is no DSNT500I message, the error is an unexpected internal error and you should consult your system programmer.

Operator response: An operator with SYSOPR authority can issue a TERM UTIL to free any held resources if that is necessary.

System programmer response: Refer to *DB2 Command Reference* for information about the TERM UTIL command. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNU316I *csect-name* **- UNABLE TO DRAIN WRITERS FROM** *object-type object-name* **PART** *part-num*

Explanation: The utility was unable to drain the class of writers from a partition of an object because the resource was unavailable. The partition is identified by object type (object-type), object name (object-name), and partition number (part-num). Message DSNT500I tells why the resource was unavailable.

Severity: 8 (error)

System action: Utility processing terminates.

Problem determination: Refer to message DSNT500I.

DSNU317I *csect-name* **DISCARD LIMIT HAS BEEN REACHED**

Explanation: The DISCARDS limit specified on the LOAD utility statement was reached.

Severity: 8 (error)

System action: Processing abends to allow users to increase the discard limit and restart the utility.

User response: Increase the discard limit and restart the utility. Otherwise -TERM the utility. Review the file of input records provided to LOAD and review table definitions to determine why the number of records discarded exceeds expectations.

Problem determination: See user response.

DSNU318I *csect-name* **RECORD 'n' WILL BE DISCARDED DUE TO INVALID** *field-name* **COLUMN SPECIFICATION FOR** *table-name*

Explanation: The load utility calculated the location of an input data value using the field specifications supplied and VARCHAR length fields present within the input data set record 'n'. The calculated position for the field was determined to be incorrect.

Severity: 4 (warning)

System action: The input data set record number is noted and will be discarded in the DISCARD phase. The LOAD utility continues to process additional records in the input data set.

User response: Following the completion of the current LOAD utility invocation, examine the record in the DISCARDS data set. Correct the error in the discarded input data, if one exists. Otherwise, correct the input field specification. Submit a RESUME(YES) LOAD invocation, which uses the DISCARDS data set and the verified LOAD utility statement and field specifications. Refer to *DB2 Utility Guide and Reference* for a definition of LOAD input field column specifications.

DSNU319I *csect-name* **RESUME(YES) WAS SPECIFIED FOR EMPTY PARTITION** *nn***. RESUME(NO) IS ASSUMED**

Explanation: This message is issued in response to a LOAD utility request containing a partition level RESUME(YES) option. The RESUME(YES) option implies that the specified partition 'nn' is populated, but in this case it was not. The message alerts the invoker to this fact, and continues as if RESUME(YES) was not specified.

Severity: 4 (warning)

System action: The LOAD utility, in attempting to honor the RESUME(YES), found the partition was empty, ignored the option, and continued as if RESUME(NO) was specified.

User response: The user is alerted to the fact that the requested partition was empty rather than populated as the RESUME LOAD option implied.

DSNU320I *csect-name* **RESUME(YES) WAS SPECIFIED FOR AN EMPTY TABLESPACE**

Explanation: RESUME(YES) was specified within the LOAD utility statement. However, the RESUME(YES) option implies that the specified table space is populated.

Severity: 4 (warning)

System action: Processing continues. The LOAD utility, in attempting to honor the RESUME(YES), found the table space empty, so ignored the option, and proceeded as if RESUME(NO) was specified. If this table is currently defined as COMPRESS YES, a dictionary is not built, unless the tablespace is empty, in which case a compression dictionary will be built.

User response: The user is alerted to the fact that the requested table space was empty rather than populated as the RESUME LOAD option implied. Use the RESUME(NO) option if you want a compression dictionary built.

DSNU321I *csect-name* **CHECK PENDING ON TABLESPACE** *tablespace-name* **PROHIBITS PROCESSING**

Explanation: The requested utility cannot be run against a tablespace or tablespace partition in a check pending state.

Severity: 8 (error)

System action: Processing terminates after the check pending states of all target tablespaces or partitions of the utility are analyzed.

User response: Remove the named tablespace or partition from its check pending state by using the CHECK DATA or REPAIR utility. If you want more information on the extent of questionable data in the table space, query the SYSTABLES and SYSTABLEPART catalog tables.

DSNU322I *csect-name* **DEFERRED STATUS OF TABLE SPACE** *database-name.table-space-name* **PROHIBITS PROCESSING**

Explanation: The requested utility cannot be run against a table space or table space partition in a deferred state.

Severity: 8 (error)

System action: Processing terminates after the deferred state of all target table spaces or table space partitions of the utility are analyzed.

User response: To remove the deferred state, use the RECOVER TABLESPACE, LOAD REPLACE utility, or -START DATABASE command with ACCESS(RW) or ACCESS(RO). If you want more information on the status of the table space, use the -DISPLAY DATABASE command.

DSNU323I *csect-name* **COLUMN** *ccc* **IS OMITTED**

Explanation: The column 'ccc' is a nonnullable column and is not being loaded. Field specifications must include all nonnullable columns. The identified column name was missing from the field specification.

Severity: 8 (error)

System action: Further utility processing is terminated.

User response: Determine if the input data contains the nonnullable column as a field. If it does, change the field specification to include it. If the input data does not contain the nonnullable column as a field, determine which is in error, the input data or the column specification. Make the corresponding correction. Resubmit the job if that is still appropriate.

System programmer response: Corrective procedures may require that a table be dropped and created again. Refer to Part 2 (Volume 1) of *DB2 Administration Guide* for a discussion of how this can be accomplished. Refer to *DB2 Utility Guide and Reference* for a discussion of field specification constraints with regard to nonnullable columns.

DSNU324I *csect-name* **DEFERRED STATUS OF INDEX SPACE** *creator-name.index-name* **PROHIBITS PROCESSING**

Explanation: The requested utility cannot be run against an index space or index space partition in a deferred state.

Severity: 8 (error)

System action: Processing terminates after the deferred state of all target index spaces or index partitions of the utility are analyzed.

User response: The deferred state can be removed through recovery or rebuild of the index space or the physical index partition. It can also be removed by the -START DATABASE command with ACCESS(RW) or ACCESS(RO). The RECOVER or REBUILD INDEX utility can be run at either the index space or index partition level. If more information on the status of the index space is required, use the -DISPLAY DATABASE command.

DSNU325I *csect-name* **- AN INVALID SQL/DS FORMAT RECORD WAS ENCOUNTERED**

Explanation: The LOAD utility detected an invalid SQL/DS format record type.

Severity: 8 (error)

System action: Utility execution is terminated.

User response: Correct the input data and resubmit the job. Data that has been transmitted from an SQL/DS unload disk data set in a VM/370 system may have extra block descriptors and record descriptors at the beginning of each record. Ensure that this situation does not exist.

Problem determination: Refer to Part 2 (Volume 1) of *DB2 Administration Guide* for additional information regarding loading data from SQL/DS and resubmitting a partially completed load step.

DSNU326I *csect-name* **- USER NOT AUTHORIZED FOR LOAD REPLACE ON TABLESPACE**

Explanation: The user attempted to use the REPLACE option of LOAD, and it will reset to empty all tables in the table space before loading data. This option requires load authorization on each table in the table space, but the user did not have authority for one or more tables. In addition, if a table in the table space has a security label column, the user must have the write-down privilege when write-down is in effect.

Severity: 12 (severe error)

System action: When LOAD REPLACE validation encounters the first nonauthorized table in the table space, the job step is terminated.

User response: Obtaining the LOAD on DATABASE privilege for the database will allow you to rev="tech" proceed using the REPLACE option. If a table in the table space has a security label column, ensure that the user has write-down privilege when write-down is in effect before using the REPLAC Eoption. The other alternative is to first delete all rows from the tables which are to be loaded, and then use the RESUME option instead of the REPLACE option.

DSNU327I *csect-name* **TABLE** *b* **IS NOT IN SAME TABLESPACE AS TABLE** *a*

Explanation: All tables that are the subject of a single LOAD utility statement must belong to the same table space.

Severity: 8 (error)

System action: Further utility processing is terminated.

User response: Remove the INTO TABLE control statement that specifies TABLE 'b' from the current LOAD request. If the data is to be loaded into TABLE 'b', create another LOAD utility statement that contains the INTO TABLE control statement for TABLE 'b' and resubmit the job.

DSNU328I *csect-name* **CANNOT PROCESS THE NON-EMPTY PARTITION** *nn* **WITH RESUME(NO). PROCESSING TERMINATES**

Explanation: The partition level RESUME keyword either specified or defaulted to a value of NO, which requires the partition to be empty. However, the LOAD command named partition 'nn', which was not empty.

Severity: 8 (error)

System action: Further utility processing is terminated.

User response: Determine why the assumption of an empty partition was wrong. This may require

consultation with the system programmer. Either remove the current partition's rows using an SQL DELETE, or change the LOAD option to RESUME YES. Resubmit the job. Refer to *DB2 Utility Guide and Reference* for a description of the partition-level RESUME option of LOAD.

System programmer response: Determine whether the partition should be empty. If it should, use SQL DELETE to empty it. Otherwise, instruct the user to use the RESUME(YES) option for that partition.

DSNU329I *csect-name* **- FIELD** *field-name* **IS NOT DEFAULTABLE**

Explanation: You specified a NULLIF or DEFAULTIF field specification option for:
- A column defined to be not nullable or not defaultable
- A field with no column counterpart.

Severity: 8 (error)

System action: The utility abends to allow restart.

User response: Determine whether the field specification refers to the correct column. If it does, remove the NULLIF option from the field specification, and restart the job. Alternatively, TERM the utility. Refer to *DB2 Utility Guide and Reference* for LOAD NULLIF option constraints.

DSNU330I *csect-name* **TABLESPACE IS NOT EMPTY**

Explanation: The RESUME keyword either specified or defaulted to a value of 'NO', which requires an empty table space. However, the LOAD command named a table space that was not empty.

Severity: 8 (error)

System action: Further utility processing is terminated.

User response: Determine why the assumption of an empty table space was wrong. This may require consultation with the system programmer. Either remove the current table(s) rows using an SQL DELETE, or change the LOAD option to RESUME YES. Resubmit the job. Refer to *DB2 Utility Guide and Reference* for a definition of the RESUME option of LOAD.

System programmer response: If the table space is used by more than one application, determine whether the table space should be empty. If it should be, use SQL DELETE to make it empty. Otherwise, instruct the user to use the RESUME(YES) option.

DSNU333I *csect-name* **- INPUT VALUE 'ebcdic-data' OUT OF RANGE FOR COLUMN 'column-name'**

Explanation: The LOAD utility detected a numeric value outside the range of permissible values for the column in the DB2 table. A loss of significant digits results if the LOAD process continues. The input data type is INTEGER EXTERNAL, DECIMAL EXTERNAL, or FLOAT EXTERNAL.

Severity: 8 (error)

System action: Further utility processing is terminated.

User response: Correct the input data and resubmit the job. Make sure that the 'field specification' on the LOAD statement is consistent with the table definition and the input data. This might mean changing:
- The table definition
- The LOAD statement
- The input data.

Refer to *DB2 Utility Guide and Reference* to determine input constraints. Refer to Part 2 (Volume 1) of *DB2 Administration Guide* for information on resubmitting a partially completed LOAD step.

DSNU334I *csect-name* **INPUT FIELD** *ffffffff* **INVALID FOR** *ttttttt*, **ERROR CODE** *cc*

Explanation: The LOAD utility found the input data provided for field *ffffffff* in table *ttttttt* to be incorrect. The error code *cc*, if present, identifies the problem. Possible error codes are:

01 Input numeric value is out of the range for the table column.

02 Input numeric field contains invalid numeric data.

03 There was an error converting input numeric data to floating-point data.

04 Input packed decimal data is in error.

05 Input is too long for a date column.

06 Input is too short for a date column.

07 Input is too long for a time column.

08 Input is too short for a time column.

09 Input is too long for a time stamp column.

10 Input is too short for a time stamp column.

11 Input character field is too long for the table column.

12 Input variable character field is too long for the table column.

13 Input is out of range for the date/time column.

14 Input is invalid for the date/time column.

15 Input length is incorrect for numeric column.

16 Input contains a blank character.

17 Input is too long for a graphic column.

18 Input is invalid for a graphic column.

19 Input is too long for the table column after a required CCSID conversion.

20 Input contained an invalid code point for a required CCSID conversion.

21 Input does not conform to the rules for valid data for a required CCSID conversion.

22 Input causes an error during CCSID conversion. An example of this type of error would be an error in a user exit routine.

23 Input is invalid for a *rowid* column.

24 Input contained a code point that when converted from the source CCSID to the target CCSID resulted in a substitution character being placed in the target string. This condition can occur when the NOSUBS keyword was specified and no discard data set was specified.

25 The range of values for an identity column is exhausted.

26 A null value has been supplied for a not null column.

27 Input is invalid for a security label column.

28 Input is a valid security label, but the seclabel and the user's seclabel are disjoint. The user does not have the authority to load this seclabel.

Severity: 8 (error)

System action: Further utility processing is terminated.

User response: Correct the input data and resubmit the job. Make sure that the 'field specification' on the LOAD statement is consistent with the table definition and the input data. This might mean changing:
- The table definition
- The LOAD statement
- The input data.

Refer to *DB2 Utility Guide and Reference* to determine input constraints. Refer to Part 2 (Volume 1) of *DB2 Administration Guide* for information on resubmitting a partially completed LOAD step.

DSNU335I *csect-name* **INPUT FIELD 'ffffffff' NOT ENTIRELY WITHIN INPUT RECORD**

Explanation: The LOAD utility found that the input field 'ffffffff' extends beyond the end of the input record.

If DISCARDS processing is active, the message severity is 4 (warning).

Severity: 4 (warning) or 8 (error)

System action: For fixed length records, with no VARCHARs included, the length needed by the LOAD utility can be calculated before processing. In this case, when the LOAD utility determines that no records can be loaded because ALL of the INTO TABLE statements require input longer than the defined lengths, the LOAD job terminates on the first record without discarding any records. Further utility processing with mixed or non-fixed length records is terminated unless DISCARDS processing is active.

User response: Correct the input data and resubmit the job. Make sure that the 'field specification' on the LOAD statement is consistent with the table definition and the input data. This might mean changing:
- The table definition
- The LOAD statement
- The input data.

Refer to *DB2 Utility Guide and Reference* to determine input constraints. Refer to Part 2 (Volume 1) of *DB2 Administration Guide* for information on resubmitting a partially completed LOAD step.

DSNU336I *csect-name* **INVALID nullif SPECIFICATION FOR INPUT FIELD ffffffff**

Explanation: The LOAD utility found the 'nullif' field specification to be incorrect.

Severity: 8 (error)

System action: Further utility processing is terminated.

User response: Correct the 'nullif' specification and resubmit the job. Refer to *DB2 Utility Guide and Reference* for a description of 'nullif' field selection options and Part 2 (Volume 1) of *DB2 Administration Guide* for information on resubmitting a partially completed LOAD step.

DSNU337I *csect-name* **RECORD(n), CONCATENATED CONTINUEIF RECORD IS TOO LARGE**

Explanation: The CONTINUEIF option was specified during LOAD to concatenate multiple physical records into one logical record for loading. The resultant concatenated record is too large to be processed. The maximum length allowed is 32767 bytes.

Severity: 8

System action: When discard processing is specified, the input data set logical record is noted and will be discarded in the DISCARD phase. The LOAD utility continues to process additional records in the input data set, and the severity level is 4. When discard process is not specified, the severity is 8 and the LOAD utility will terminate.

User response: When discard processing is specified, following the completion of the current LOAD utility invocation, examine the record in the DISCARDS data set. Correct the error in the discarded input data, and submit a RESUME(YES) LOAD invocation with the DISCARDS data set as the input data set. When discard processing is not specified: correct the error in the input data set and rerun the job.

System programmer response: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: Collect the following diagnostic items: Appendix A, "Problem determination," on page 659: 1, 2, 5.

DSNU338I *csect-name* **INVALID COLUMN SPECIFICATION FOR** *column name*

Explanation: The LOAD utility found the dynamically determined column location for an input field to be incorrect.

Severity: 8 (error)

System action: The utility abends to allow restart.

User response: Correct the input field specification or the input data and restart the job. Alternatively, TERM the utility.

DSNU339I *csect-name* **INVALID CONTINUEIF SPECIFICATION**

Explanation: The CONTINUEIF clause of a LOAD utility extends beyond the end of the input record.

Severity: 8 (error)

System action: The LOAD utility is terminated.

User response: Correct the CONTINUEIF clause. The column specification must be reduced and/or the value being compared must be shortened. The value being compared must be entirely within the input record.

DSNU340I *csect-name* **- ERROR LOADING INDEX, DUPLICATE KEY, INDEX=***index-name***, TABLE=***table-name***, RID OF INDEXED ROW=***rid-of-indexed-row***, RID OF NON-INDEXED ROW=***rid-of-non-indexed-row***

Explanation: A duplicate key was found while building the indicated index during the BUILD phase of the REORG or REBUILD INDEX utility. This violates the uniqueness rule for that index.

Severity: 8 (error)

System action: All duplicate index entries of the indicated index are reported and the index is in RECOVERY PENDING status. Duplicate rows are present in the indexed table.

Utility processing continues with any other utility functions within the utility job step.

User response: Duplicate rows are present in the indexed table. The system programmer should be informed, so that any applications that are dependent upon the uniqueness of the table rows are not initiated until corrective action has been taken.

Operator response: Notify the system programmer.

System programmer response: Steps should be taken to inhibit any further processing until the index is repaired. If the nonunique keys are correct, DROP and recreate the index as nonunique. If the records with duplicate keys should not be in the table, use REPAIR DELETE to remove the erroneous rows. Use the RID values provided in this message as input to the REPAIR utility. After the duplicate data rows are removed or the unique index has been dropped, issue the -DISPLAY DATABASE command with RESTRICT against the target database. Use the REBUILD INDEX utility to recover any indexes that are in the REBUILD pending state. CHECK INDEX can be run to verify that the data and indexes are consistent.

Problem determination: Use REPAIR LOCATE RID DUMP to view the records with duplicate keys. SQLSELECT may fail due to the inconsistent index. Use CHECK INDEX to check the validity of the index.

DSNU341I *csect-name* **- IMAGE COPY MUST BE TAKEN BEFORE REORG CAN BE RUN ON A DB2 CATALOG OR DIRECTORY TABLE SPACE**

Explanation: The DB2 catalog or directory table space being reorganized does not have an image copy that the RECOVER utility can use to recover the table space to currency. Before running the REORG utility, you must run the COPY utility to create an image copy. This prevents having to recover the DB2 catalog or directory table space to a prior point in time if a problem occurs during REORG processing.

Severity: 8

System action: Processing terminates

User response: Run the COPY utility to create an image copy of the DB2 catalog or directory table space. If more than one catalog or directory table space is to be reorganized, consider using the QUIESCE and COPY utilities for the set of table spaces to be reorganized.

Use the REPORT utility to view the recovery information related to the table space. The output from the REPORT utility should indicate either the absence of an image copy, or the presence of a REORG LOG NO SYSCOPY record.

DSNU342I *csect-name* **ERROR LOADING INDEX, ERROR = X***reason-code***, INDEX =** *index-name*

Explanation: The LOAD, REORG or REBUILD INDEX utility encountered a processing error, which it could not correct, while extending the index identified by 'index-name'. Refer to *DB2 Codes* for an explanation of the code given in 'reason-code'.

Severity: 8 (error)

System action: The building of index entries for the indicated index is not completed. An SVC dump of the database services address space is requested for diagnostic purposes. The extension of other indexes associated with the tables being loaded is attempted. Any subsequent attempt to use the index directly or indirectly will cause the invoker to abend.

User response: If the indexed table is used by more than one application, the system programmer should be informed so that applications that insert rows into, or delete rows from, the table or use the index directly are inhibited until corrective action is taken.

Normal corrective action would be to use the REBUILD INDEX utility option to reconstruct the index after the processing problem identified by the reason code has been resolved. Alternatively, the index can be dropped and then created again, unless the index in question is the partitioning index.

Operator response: The table space in question might need to be stopped and started for utility-only processing to inhibit attempted application use of the defective index.

System programmer response: If the index is used by more than one application, applications that either insert rows into, or delete rows from, the indexed table or use the index to access rows must be inhibited from executing. One way to accomplish this is to place the table space in question under utility only processing control via the -START and -STOP DATABASE commands until the REBUILD INDEX utility is executed.

Alternatively, the index can be dropped. Application execution can then proceed, and the index can be created again (unless the index was the partitioning index). If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem. Refer to *DB2 Codes* to determine the processing error indicated by the reason code. For a discussion of how a table can be dropped and created again, or what to do if a partitioning index is damaged, refer to the discussion of LOAD processing in Part 2 (Volume 1) of *DB2 Administration Guide*.

DSNU343I *csect-name* **- BUILD PHASE STATISTICS.** *nnn* **DUPLICATE KEY ERRORS ENCOUNTERED**

Explanation: The count of duplicate key errors encountered during processing is 'nnn'.

Severity: 4 (warning) or 8 (error)

System action: Processing continues. For the LOAD utility, the records causing the duplicate key errors will be deleted in a subsequent phase.

DSNU344I *csect-name* **- UNIQUE INDEX KEY DUPLICATES KEY OF INDEXED ROW AT RID** *rrrr* **INDEX =** *index-name* **TABLE =** *table-name* **RECNO =** *input-dataset-record-number* **RID =** *duprid*

Explanation: The input data set record indicated was loaded into table *table-name* at RID *duprid*, but it causes a unique index violation for index *index-name*. It duplicates the key value of a record already in the table at RID *rrrr*.

Severity: 4 (warning)

System action: Processing continues. No index entry is created for the record at RID *duprid*. The record will be deleted in a subsequent phase of processing.

DSNU345I *csect-name* **UNIQUE INDEX KEY DUPLICATES KEY FROM INPUT DATASET RECORD** *iiii* **LOADED AT RID** *rrrr* **INDEX =** *index-name* **TABLE =** *table-name* **RECNO =** *recno* **RID =** *duprid*

Explanation: Input data set record *recno* was loaded into table *table-name* at RID *duprid*, but it causes a unique index violation for index *index-name*. It duplicates the key value of another record in the load job: the record loaded at RID *rrrr* from input data set record *iiii*.

Severity: 4 (warning)

System action: Processing continues. Index entries are not created for either record. Both records will be deleted in a subsequent phase of processing.

DSNU346I **csect-name RECORD(n), CONCATENATED SQL/DS RECORD IS TOO LARGE**

Explanation: A LOAD utility was executing with FORMAT SQL/DS input data. Multiple spanned segments of the input data were concatenated into one logical record for loading. The resultant concatenated record is too large to be processed. The maximum length allowed is 32764 bytes.

Severity: 4 or 8

System action: When discard processing is specified, the input data set logical record is noted and is discarded in the DISCARD phase. The LOAD utility continues to process additional records in the input data set and the severity is 4. When discard processing is not specified, the severity is 8 and the LOAD utility terminates.

User response: When discard processing is specified, examine the records in the DISCARDS data set after the LOAD utility completes. Correct the error in the discarded input data. Then run the LOAD utility with the RESUME YES option using the DISCARDS data set as input. When discard processing is not specified, correct the error in the input data set and rerun the job.

System programmer response: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 5.

DSNU347I *csect-name* **DEADLINE =** *yyyy-mm-dd-hh.mm.ss.mmmmmm*

Explanation: This message applies only to REORG with SHRLEVEL REFERENCE or CHANGE. It supplements message DSNU105I, which can appear for any utility (including REORG).

This message is issued in response to a -DISPLAY UTILITY command.

It is also issued by the REORG utility if the user specified SHRLEVEL REFERENCE or CHANGE and DB2 estimates that the switch phase will not start by the deadline specified by the DEADLINE phrase.

This message indicates the most recently specified values of the DEADLINE parameter to REORG. For this parameter, a previous invocation of the -ALTER UTILITY command might have specified a value. If no previous invocation of -ALTER UTILITY has specified a value, then the most recently specified value is the value that was specified with the invocation of the REORG utility.

yyyy-mm-dd-hh.mm.ss.mmmmmm indicates a timestamp, according to the most recently specified value of DEADLINE.

Severity: 0 (informational)

System action: The message reflects the state of DB2 at the time of the -DISPLAY UTILITY command or the time when DB2 *estimated* that the switch phase would not start by the deadline specified by the DEADLINE phrase.

If this message was issued for -DISPLAY UTILITY, the system continues processing. If an -ALTER UTILITY command is then executed, a subsequent -DISPLAY UTILITY command would reflect the results (if any) of the -ALTER UTILITY command.

DSNU348I *csect-name* **BUILD PHASE STATISTICS. NUMBER OF KEYS =** *nnnnnnnn* **FOR INDEX** *index-name* **PART** *part-num*

Explanation: This message is issued by the BUILD phase of the REBUILD INDEX utility to report the total number of index key entries used to build the indicated index partition.

Severity: 0 (informational)

System action: This message indicates the number of index key entries submitted to the DB2 data manager to build the indicated index partition. Both unique and nonunique key entries are counted. Rejected key entries are not counted.

User response: The user can verify that the number reported is consistent with expectations.

DSNU349I *csect-name* **BUILD PHASE STATISTICS. NUMBER OF KEYS =** *nnnnnnnn* **FOR** *index-name*

Explanation: This message is issued by the BUILD phase of the LOAD, REORG or REBUILD INDEX utility. The message is repeated for each index built by the LOAD, REORG or REBUILD INDEX utility. It indicates the total number of index key entries used to build the indicated index.

Severity: 0 (informational)

System action: This message indicates the number of index key entries that were submitted to the DB2 data manager to build the indicated index. Both unique and nonunique key entries are counted. Rejected key entries are not counted.

User response: The user can verify that the number reported is consistent with expectations. For example, the user can determine if it matches the number of records loaded for the table indexed by the indicated index.

DSNU350I *csect-name* **- EXISTING RECORDS DELETED FROM TABLESPACE**

Explanation: A LOAD utility execution, in which REPLACE was specified, has deleted all records from the specified table space.

Severity: 0 (informational)

System action: Processing continues.

DSNU351I *csect-name* **- '***reorg-option***' OPTION IS NOT APPLICABLE FOR A DB2 CATALOG OR DIRECTORY OBJECT. THE OPTION IS IGNORED.**

Explanation: The identified option does not apply for this DB2 catalog or directory object. Table spaces and indexes within the DB2 catalog and directory may be processed differently than other table spaces and indexes. Consequently, these table spaces and indexes may not make use of reorg option *reorg-option*.

Severity: 4 (warning)

System action: The option is ignored and processing continues.

User response: None is required. However, you can remove the option from the utility input statement.

DSNU352I EXISTING RECORDS DELETED FROM TABLESPACE PARTITION x

Explanation: A LOAD utility execution, in which PART 'x' REPLACE was specified, has deleted all data records from partition 'x'. Index records over partition 'x' will be deleted in a subsequent phase of processing.

Severity: 0 (informational)

System action: Processing continues.

DSNU353I *csect-name* **RECORD '***n***' WILL BE DISCARDED DUE TO CHECK CONSTRAINT** *constraint-name* **VIOLATION ON TABLE** *table-name*

Explanation: The discarded record violates the check constraint named *constraint-name* defined on table *table-name*. The record is not loaded and is discarded during the DISCARD phase.

Severity: 4 (warning)

System action: The LOAD utility continues to process additional records in the input data set.

User response: After the LOAD utility completes, examine the record in the DISCARDS data set. Correct the error in the discarded input data. Submit a RESUME(YES) LOAD job with the DISCARDS data set as the input data set.

DSNU354I *csect-name* **RECORD '***n***' VIOLATES CHECK CONSTRAINT** *constraint-name* **ON TABLE** *table-name*

Explanation: The LOAD utility cannot load record *n* into table *table-name* because the check constraint *constraint-name* would be violated. The record is not loaded.

Severity: 8 (error)

System action: Processing terminates. An abend is requested so that the LOAD can be restarted after the error condition is corrected.

User response: Find the failing record in the input data set, correct the condition that caused the check constraint to be violated, and restart the LOAD utility. You can optionally include a SYSDISC DD statement in the JCL input to hold copies of records that are not loaded.

DSNU355I *csect-name* **INDEXVAL PHASE STATISTICS -** *nnn* **DUPLICATE KEY ERRORS, KEY EVALUATION ERRORS, OR LOB ERRORS WERE CORRECTED BY DELETING** *rrr-num* **DATA ROWS**

Explanation: This message reports the results of the INDEXVAL phase of the LOAD utility. Data rows that were added during the LOAD or RELOAD phase of this LOAD utility invocation were deleted because the data rows caused one of the following conditions:

- Unique index violations in the BUILD phase
- Key evaluation errors during the build of an index on an expression
- LOB errors in the BUILD phase

A total of *rrr* data rows were deleted to correct *nnn* unique index violations or key evaluation errors.

Unique index violations are counted in one of two ways. An input data set record which has a value for a uniquely indexed column that duplicates a value that existed in the index prior to the LOAD job is counted as one duplicate key error. Two input data set records which have like values for a uniquely indexed column are counted as two unique index errors (that is, each of the two input data set records is in error).

Severity: 4 (warning)

System action: Processing continues.

User response: Examine the error summary from the LOAD job. Determine which of the discarded records need to be corrected and loaded into the table space.

Compare the number of duplicate key errors corrected in the INDEXVAL phase to the number of duplicate key errors detected in the BUILD phase (reported by message DSNU343I). If the counts are unequal, notify the system programmer.

If the counts are equal, following the LOAD utility execution, examine the discards file or the LOAD error summary to determine which input data set records caused the unique index violations. Revise the input records so they will not introduce duplicate values into unique indexes and reapply the input records to the table space. Alternatively, if appropriate, make the index non-unique and reapply the rejected records to the table space.

System programmer response: If you suspect an error

in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 5.

DSNU356I *csect-name* **INDEXVAL PHASE COMPLETE, ELAPSED TIME =** *hh/mm/ss*

Explanation: The INDEXVAL phase of the LOAD utility has completed normally. The elapsed time for the execution was 'hh' hours, 'mm' minutes, and 'ss' seconds.

Severity: 0 (informational)

DSNU357I *csect-name* **- 'LOG NO' OPTION REQUIRED FOR DB2 CATALOG AND DIRECTORY TABLE SPACES**

Explanation: The LOG NO option must be specified when reorganizing a DB2 catalog or directory table space.

Severity: 8 (error)

System action: Processing terminates.

User response: Add the LOG NO option to the utility input statement.

DSNU358I *csect-name* **- 'UNLOAD ONLY' OPTION NOT VALID FOR DB2 CATALOG OR DIRECTORY TABLE SPACE**

Explanation: The UNLOAD ONLY option is not permitted for DB2 catalog or directory table spaces.

Severity: 8

System action: Processing terminates.

User response: Determine the purpose for using the UNLOAD ONLY option. There might be alternative ways of accomplishing the required function (for example, the UNLOAD PAUSE option).

DSNU359I **KEYWORD 'SORTKEYS' SPECIFIED BUT NO INDEX OR FOREIGN KEYS EXIST, KEYWORD IS IGNORED**

Explanation: The 'SORTKEYS' keyword was specified for a LOAD or REORG utility job; however, it was found that no index and no foreign keys were defined for any of the tables being loaded or reorganized. The keyword is ignored and processing continues normally.

Severity: 0 (informational)

System action: Processing continues normally.

User response: If an index or foreign keys were expected to be found for the table or tables being loaded or reorganized, determine the reason the index or foreign keys could not be found.

Problem determination: Probably a user error. Determine if an index should be defined on any of the tables in the table space, or if foreign keys should be defined. If so, define the index or foreign keys; otherwise, remove the 'SORTKEYS' keyword from the utility statement if necessary.

DSNU360I *csect-name* **- IN REORG WITH SHRLEVEL REFERENCE OR CHANGE, DATA SET** *name* **EXISTS, CONTEXT=***c*

Explanation: For the area (table space or partition) being reorganized, REORG with SHRLEVEL REFERENCE or CHANGE unloads data from the original data sets and reloads data into shadow data sets. REORG then exchanges the names of the original and shadow data sets, using temporary names during the renaming. The original name of a data set is in the form "*catname*.DSNDB*x*.*dbname*.*psname*.I0001.A*nnn*". The shadow name of a data set is in the form "*catname*.DSNDB*x*.*dbname*.*psname*.J0001.A*nnn*". The temporary name of a data set is in the form "*catname*.DSNDB*x*.*dbname*.*psname*.T0001.A*nnn*". REORG uses such data sets for indexes as well as data.

During its initialization, REORG cannot execute if:

- Data sets with the temporary names already exist.
- DB2-managed data sets with the shadow names already exist.
- User-managed data sets with the shadow names do **not** already exist.

To exchange the names of the original and shadow data sets, REORG performs the following renaming:

1 In the switch phase, it renames the original data set from the original name to the temporary name

2 In the switch phase, it renames the shadow data set from the shadow name to the original name

3 In the utilterm phase, it renames the original data set from the temporary name to the shadow name (for user-managed data sets). It deletes the temporary for DB2-managed data sets.

If a -TERM UTILITY command is issued after any renaming, or a restart of the REORG switch phase occurs after any renaming, the data set names revert back to their original form prior to the renaming:

4 The original data set from the shadow name to the temporary name

5 The shadow data set from the original name to the shadow name

6 The original data set from the temporary name to the original name

The context code indicates the specific type of error:

1 During the switch phase of REORG, on rename 1, REORG detected that a data set with a temporary name already exists. This error condition can arise for a DB2-managed or user-managed data set.

2 During the switch phase of REORG, on rename 2, REORG detected that a data set with a original name already exists. This error condition can arise for a DB2-managed or user-managed data set.

3 During the utilterm phase of REORG, on rename 3, REORG detected that a data set with a shadow name already exists. This error condition can arise for a user-managed data set.

4 During switch phase restart or a -TERM UTILITY during the switch phase, on rename 4 (while reverting back to the original name), -TERM UTILITY detected that a data set with a temporary name already exists. This error condition can arise for a DB2-managed or user-managed data set.

5 During switch phase restart or a -TERM UTILITY during the switch phase, on rename 5 (while reverting back to the original name), -TERM UTILITY detected that a data set with a shadow name already exists. This error condition can arise for a DB2-managed or user-managed data set.

6 During switch phase restart or a -TERM UTILITY during the switch phase, on rename 6 (while reverting back to the original name), -TERM UTILITY detected that a data set with a original name already exists. This error condition can arise for a DB2-managed or user-managed data set.

8 During initialization of REORG, REORG detected that a data set with a temporary name already exists. This error condition can arise for a DB2-managed or user-managed data set.

Severity: 8 (error)

System action: For context code 1, or 2, the utility job step that caused the function to be requested abends with code 00E4031B. The utility job is placed in the stopped state.

For context code 3, the utility job step that caused the function to be requested continues without performing this renaming.

For context code 4, 5, or 6, the object is left in a recovery pending state, -TERM UTILITY continues to undo renaming, and a message is issued. The objects and messages are as follows:

- If the data set is part of a nonpartitioned table space, the object is the table space, and the message is DSNU562I.
- If the data set is part of a partition of a table space, the object is the partition, and the message is DSNU560I.
- If the data set is part of a nonpartitioned index, the object is the index, and the message is DSNU557I.
- If the data set is part of a partition of a partitioned index, the object is the partition of the index, and the message is DSNU559I.

For context code 8, REORG does not execute.

User response: For context code 1 or 2, you can choose between these actions:

- Delete the data set; then RESTART the utility.
- TERM the utility. Delete the data set before trying to execute REORG again with SHRLEVEL REFERENCE or CHANGE.

For context code 3, if you want to perform the renaming, delete the data set with the shadow name and then perform the renaming.

For context code 4, 5, or 6, execute the RECOVER or REBUILD INDEX utility; then delete the shadow and temporary data sets (if they still exist).

For context code 8, delete the data set and then execute REORG.

DSNU361I *csect-name* **FOREIGN KEY HAS NO PRIMARY KEY FOR RELATIONSHIP=***r1* **INPUT RECORD NUMBER=***nnnn***, RID=X***rrrr*

Explanation: Input record *nnnn* caused a row to be loaded into table *table* at RID *rrrr* with no primary key for the relationship with the name *r1*.

Severity: 4 (warning)

System action: Processing continues.

User response: None required. The constraint violation described will be rectified later in processing.

DSNU362I *csect-name* **- IN REORG WITH SHRLEVEL CHANGE, LOG HAS BECOME SHORT, BUT MAXRO IS DEFER, MEMBER=***member-name***, UTILID=***utilid*

Explanation: In execution of the REORG utility with SHRLEVEL CHANGE, if the value of MAXRO is DEFER, and the actual time for an iteration and the estimated time for the next iteration are both less than 5 seconds, then DB2 adds a 5 second pause to the next

iteration. This pause limits the utility's consumption of processor time.

Within an execution of REORG, the first time that this situation occurs, DB2 produces this message, which announces the pause. On all later occurrences of this situation, DB2 still adds the pause, but DB2 sends the message only if DB2 has *not* sent the message (for this execution of REORG) within the past 30 minutes. This is a WTO message; it is sent to the console, but it is not an action message. It is not sent to SYSPRINT.

In a data sharing environment, *member-name* identifies the DB2 member. In a non-data-sharing environment, *member-name* is blank. *utilid* identifies the utility job step.

Severity: 0 (informational)

System action: REORG continues execution, with a 5 second pause. If the conditions in the explanation above continue to be true, DB2 will continue adding 5 second pauses.

User response: If you have been planning to wait for an appropriate time to execute the -ALTER UTILITY command to choose the value of MAXRO and thus cause REORG to finish, now might be an appropriate time. However, you are not required to take any action.

Operator response: If you have been planning to wait for an appropriate time to execute the -ALTER UTILITY command to increase the value of MAXRO and thus cause REORG to finish, now might be an appropriate time. However, you are not required to take any action.

DSNU363I *csect-name* **- ENFORCE PHASE STATISTICS - RELATIONSHIP** *r1* **FOREIGN KEYS CHECKED=***n* **CONSTRAINT VIOLATIONS ENCOUNTERED=***m*

Explanation: This message provides statistics for the enforce phase of the load utility. For relationship 'r1', the number of foreign keys checked is 'n'. The number of constraint violations found is 'm'.

Severity: 0 (informational)

System action: Processing continues.

DSNU364I csect-name PARTITIONS WILL BE LOADED IN PARALLEL, NUMBER OF TASKS = nnnn

Explanation: This message is issued by the RELOAD phase of the LOAD utility. The message indicates that the LOAD utility is able to load partitions in parallel. The number of utility tasks used to load partitions is indicated by *nnnn*.

Severity: 0

System action: Processing of the RELOAD phase continues.

User response: You can verify that the number of tasks reported is consistent with your expectations. For example, if 4 partitions are to be loaded, you might expect to see the number of tasks = 4.

DSNU365I *csect-name* **- USER** *auth-id* **DOES NOT HAVE AUTHORITY ON REORG MAPPING TABLE**

Explanation: The REORG utility was run with SHRLEVEL CHANGE and a mapping table, but user *auth-id* does not have authority to delete, insert, select, and update in the mapping table specified in the utility statement.

Severity: 8 (error)

System action: Further utility processing is terminated.

User response: Obtain authorization to delete, insert, select, and update in the mapping table specified in the utility statement.

DSNU366I *csect-name* **- REORG MAPPING TABLE HAS WRONG COLUMNS**

Explanation: A mapping table was specified for a utility job executing REORG with SHRLEVEL CHANGE. However, the table has the wrong set of columns. This message can arise for any parameters of CREATE TABLE that are missing from, added to, or different from the CREATE TABLE statement that appears for the mapping table in the REORG section of *DB2 Utility Guide and Reference*.

Severity: 8 (error)

System action: REORG does not execute.

User response: Create a mapping table with the correct set of columns, and no other columns, as specified under REORG in *DB2 Utility Guide and Reference*. Execute REORG.

DSNU367I *csect-name* **- ENFORCE PHASE STATISTICS -** *n* **ROWS DELETED FROM TABLE** *t*

Explanation: This message reports the results of the ENFORCE phase of the LOAD utility. Data rows added during the RELOAD phase of the LOAD utility have been deleted because they contained foreign keys which did not have a corresponding primary key. A total of *n* data rows were deleted from table *t*.

Primary errors may cascade and remove additional records. The count is the sum of both primary and secondary constraint violations.

Severity: 4 (warning)

System action: Processing continues.

User response: Determine why no primary key exists for the deleted record.

Problem determination: All primary errors are accompanied by a DSNU361I message which identifies the relationship and input record which does not have a primary key value. The report phase will report all secondary errors and indicate the primary error record which caused this record to be deleted.

The relevant parent table can be found by querying the SYSIBM.SYSRELS catalog table. The relevant columns can be found by querying the SYSIBM.SYSFOREIGNKEYS catalog table.

DSNU368I *csect-name* **- REORG MAPPING TABLE HAS WRONG INDEX**

Explanation: A mapping table was specified for a utility job executing REORG with SHRLEVEL CHANGE. However, the table has the wrong index. This message can arise for any parameters of CREATE INDEX that are missing from, added to, or different from the CREATE INDEX statement that appears for the mapping table in the REORG section of *DB2 Utility Guide and Reference*. It can also arise for the existence of any other indexes on the mapping table.

Severity: 8 (error)

System action: REORG does not execute.

User response: Create a mapping table with the correct index, and no other indexes, as specified under REORG in *DB2 Utility Guide and Reference*. Execute REORG.

DSNU369I *csect-name* **- ENFORCE PHASE COMPLETE, ELAPSED TIME =** *hh/mm/ss*

Explanation: The ENFORCE phase of the LOAD utility has completed normally. The elapsed time for the execution was 'hh' hours, 'mm' minutes, and 'ss' seconds.

Severity: 0 (informational)

DSNU370I *csect-name* **- THE LOCAL FORMAT OPTION HAS BEEN USED WITH A DATE OR TIME AND NO LOCAL EXIT HAS BEEN INSTALLED.**

Explanation: The Installation Application Programming Defaults Panel indicated that an exit for date or time was supplied, but the exit supplied by DB2 was not replaced.

Severity: 8 (error)

System action: The statement cannot be executed.

User response: Contact the system programmer about installation of the date or time exit.

System programmer response: To replace the DB2-supplied exit routine, an SMP USERMOD must be applied. If an SMP USERMOD is undesirable, the replacement module may be linked into a user library either included in the MVS LINKLIST or concatenated to the STEPLIB in the started task JCL for *ssnm*DBM1 (*ssnm* is the subsystem name).

DSNU371I *csect-name* **- REORG MAPPING TABLE'S TABLE SPACE MUST BE SEGMENTED**

Explanation: REORG with SHRLEVEL CHANGE specified a mapping table for a table space that is not segmented. The table space that contains the mapping table must be segmented.

Severity: 8 (error)

System action: REORG does not execute.

User response: Specify a mapping table that is in a table space that is segmented.

DSNU372I *csect-name* **- CANNOT REORGANIZE REORG MAPPING TABLE'S TABLE SPACE**

Explanation: REORG with SHRLEVEL CHANGE specified a mapping table that resides in the table space that is being reorganized. The table space that contains the mapping table must not be the table space that is being reorganized.

Severity: 8 (error)

System action: REORG does not execute.

User response: Specify a mapping table that is not in the table space that is being reorganized.

DSNU373I *csect-name* **- IN -ALTER UTILITY OR REORG WITH SHRLEVEL REFERENCE OR CHANGE, DEADLINE HAS ALREADY OCCURRED, CONTEXT=***c*

Explanation: If a deadline is specified in the -ALTER UTILITY command or the REORG utility with SHRLEVEL REFERENCE or CHANGE, the deadline cannot occur before the time of execution of the -ALTER UTILITY or REORG that specified the deadline. Context codes are:

1 The error occurred on -ALTER UTILITY.

2 The error occurred on REORG.

Severity: 8 (error)

System action: -ALTER UTILITY or REORG does not execute.

User response: Execute -ALTER UTILITY or REORG with no deadline or with a deadline that has not already occurred. If you want to terminate an execution of REORG, use the -TERM UTILITY command.

DSNU374I *csect-name* **- REORG WITH SHRLEVEL REFERENCE or CHANGE HAS BEEN TERMINATED, REASON=***c*

Explanation: REORG with SHRLEVEL REFERENCE or CHANGE has been terminated.

1 REORG with SHRLEVEL CHANGE specified LONGLOG TERM, and at the end of an iteration of log processing, the number of log records that the next iteration would process is not sufficiently lower than the number that the previous iterations processed.

2 DB2 estimated that the switch phase of REORG with SHRLEVEL REFERENCE or CHANGE would not start by the deadline specified in the DEADLINE phrase.

Severity: 8 (error)

System action: REORG terminates.

User response: For reason code 1, attempt a new invocation of reorganization later, perhaps when you expect that the application workload is lower or the maximum desired amount of time for applications to have read-only access is longer.

For reason code 2 on SHRLEVEL CHANGE, attempt a new invocation of reorganization later, perhaps when you expect to have a longer time available, you expect that the application workload is lower, or you expect that the maximum desired amount of time for applications to have read-only access is longer. For reason code 2 on SHRLEVEL REFERENCE, attempt a new invocation of reorganization later, perhaps when you expect to have a longer time available.

DSNU375I *csect-name* **DISCARD PHASE STATISTICS -** *nnn* **INPUT DATA SET RECORDS DISCARDED**

Explanation: Input data set records were rejected by the LOAD utility for one or several reasons. The rejected records have been placed in a discards data set. The number of records rejected is 'nnn'. Message DSNU399I summarizes the errors causing the records to be rejected.

Severity: 4 (warning)

System action: Processing continues.

User response: Examine the LOAD error summary and decide which of the discarded records, if any, should be corrected and reapplied to the table space.

DSNU376I *csect-name* **DISCARD PHASE COMPLETE, ELAPSED TIME =** *hh/mm/ss*

Explanation: The DISCARD phase of the LOAD utility has completed normally. The elapsed time for the execution was 'hh' hours, 'mm' minutes, and 'ss' seconds.

Severity: 0 (informational)

DSNU377I *csect-name* **- IN REORG WITH SHRLEVEL CHANGE, THE LOG IS BECOMING LONG, MEMBER=***member-name***, UTILID=***utilid*

Explanation: During log processing for REORG with SHRLEVEL CHANGE, the number of log records that the next iteration will process is not sufficiently lower than the number that the previous iterations processed. This situation means that during reorganization, the log is not being read quickly enough to catch up to the application writing the log. This is a WTO message; it is sent to the console, but it is not an action message. It is also sent to SYSPRINT.

DB2 refrains from taking the action specified in the LONGLOG phrase if any one of the following events occurs for this execution of REORG before the delay expires:

- An -ALTER UTILITY command is issued.
- A -TERM UTILITY command is issued.
- DB2 estimates that the time to perform the next iteration will be less than or equal to the time specified in the MAXRO phrase.
- REORG terminates for any reason (including the deadline).

member-name
Identifies the DB2 member in a data sharing environment. Blank in a non-data-sharing environment.

utilid Identifies the utility job step.

Severity: 4 (warning)

System action: After a delay (specified by the DELAY phrase of the REORG utility or the -ALTER UTILITY command), REORG performs the action specified by the LONGLOG phrase of the REORG utility or the -ALTER UTILITY command.

User response: No action is required. However, if the action specified by the LONGLOG phrase is not desired, the operator (or a user-written program that is triggered by the message) can take an action. For a list of possible actions, see the description of LONGLOG in the REORG section of *DB2 Utility Guide and Reference*.

Operator response: No action is required. However, if the action specified by the LONGLOG phrase is not desired, the operator (or a user-written program that is triggered by the message) can take an action. For a list of possible actions, see the description of LONGLOG in the REORG section of *DB2 Utility Guide and Reference*.

DSNU378I *csect-name* **- REORG WITH SHRLEVEL NONE HAS INAPPROPRIATE NOSYSREC KEYWORD, REASON=***c*

Explanation: REORG with SHRLEVEL NONE allows the following combinations of the SORTDATA and NOSYSREC keywords:
- Neither keyword
- SORTDATA alone (without NOSYSREC)
- SORTDATA NOSYSREC

These combinations do not trigger this message. However, NOSYSREC alone (without SORTDATA) is inappropriate and triggers this message with a reason code of 1.

Also, for REORG with SHRLEVEL NONE, if you specify UNLOAD PAUSE, UNLOAD ONLY, or UNLOAD EXTERNAL, NOSYSREC is inappropriate and triggers this message with a reason code of 2. If the two conditions (omission of SORTDATA and inclusion of UNLOAD PAUSE, ONLY, or EXTERNAL) are both true, NOSYSREC triggers this message with a reason code of 2.

Severity: 8 (error)

System action: REORG does not execute.

User response: Specify an invocation of REORG with appropriate parameters. Specifically, for reason code 1, if you want to execute REORG with SHRLEVEL NONE, add SORTDATA and/or remove NOSYSREC. For reason code 2, if you want to execute REORG with SHRLEVEL NONE, remove NOSYSREC, UNLOAD PAUSE, UNLOAD ONLY, or UNLOAD EXTERNAL .

DSNU379I *csect-name* **- ALTER UTILITY, UTILID =** *utilid***: SWITCH PHASE IN REORG HAS ALREADY STARTED, NO ALTER IS PERFORMED**

Explanation: The -ALTER UTILITY command was applied to a job that is executing REORG with SHRLEVEL REFERENCE or CHANGE. However, nothing was altered, because REORG had already started its switch phase.

Severity: 4 (warning)

System action: ALTER UTILITY has no effect.

DSNU380I *csect-name* **TABLESPACE** *database.tablespace-name* **PARTITION** *part-num* **IS IN COPY PENDING**

Explanation: An image copy is required to establish a new base from which the table space partition can be recovered. Otherwise, the results of running the RECOVER utility on that table space partition are unpredictable.

If the RECOVER TABLESPACE utility with the keyword TOCOPY is being used, the START_RBA or START_LRSN value of the associated full image copy might be outside the available recovery information. To verify this, look at the REPORT utility output. The RECOVER TABLESPACE utility will process the TOCOPY keyword and put the recovering object in copy pending status.

Severity: 4 (warning)

User response: Take an image copy of the associated table space partition. If the message was issued after the COPY utility failed, correct all specified errors for the associated table space before attempting another image copy of that table space.

DSNU381I *csect-name* **TABLESPACE** *database.tablespace-name* **IS IN COPY PENDING**

Explanation: This message is issued after:
- A LOAD or REORG utility with the LOG(NO) option left the associated table space in an unrecoverable state.
- An image copy failed because broken pages were detected or an image copy was terminated.
- A partial recovery finished successfully.
- A MODIFY RECOVERY left the associated table space in an unrecoverable state.

  If the RECOVER TABLESPACE utility with keyword TOCOPY is being used, the START_RBA or START_LRSN value of the associated full image copy might be outside the available recovery information. To verify this, look at the REPORT utility output. The RECOVER TABLESPACE utililty will process the TOCOPY keyword and put the recovering object in copy pending status.

An image copy is required to establish a new base from which the table space can be recovered. Otherwise, the results of running the RECOVER utility on that table space are unpredictable.

Severity: 4 (warning)

User response: Take an image copy of the associated table space. If the message was issued after the COPY utility failed, correct all specified errors for the associated table space before attempting another image copy on that table space.

DSNU382I *csect-name* **- REORG UTILITY, UTILID =** *utilid* **REQUESTED TO ALTER, PHASE =** *utility-phase*

Explanation: This message is issued in response to an -ALTER UTILITY command. It acknowledges that the request for alteration was received and verified. At the time the -ALTER UTILITY request was received, REORG was in the utility job step identified by UTILID *utilid*.

Severity: 0 (informational)

System action: DB2 has accepted the request to alter the indicated utility job step. The utility alters its parameter values.

User response: This message is an acknowledgement that the request has been accepted and is in progress.

DSNU383I *csect-name* **CURRENT ITERATION NUMBER =** *nnnnnn* **WRITE ACCESS ALLOWED IN CURRENT ITERATION =** *www* **ITERATION BEFORE PREVIOUS ITERATION: ELAPSED TIME =** *ha:ma:sa* **NUMBER OF LOG RECORDS PROCESSED =** *nnnnna* **PREVIOUS ITERATION: ELAPSED TIME =** *hb:mb:sb* **NUMBER OF LOG RECORDS PROCESSED =** *nnnnnb* **CURRENT ITERATION: ESTIMATED ELAPSED TIME =** *hc:mc:sc* **ACTUAL ELAPSED TIME SO FAR =** *hd:md:sd* **ACTUAL NUMBER OF LOG RECORDS BEING PROCESSED =** *nnnnnd* **CURRENT ESTIMATE FOR NEXT ITERATION: ELAPSED TIME =** *he:me:se* **NUMBER OF LOG RECORDS TO BE PROCESSED =** *nnnnne*

Explanation: This message applies only during the LOG phase of REORG with SHRLEVEL CHANGE. It supplements message DSNU105I which can appear in any phase (including the LOG phase).

This message is issued in response to a -DISPLAY UTILITY command.

It is also issued by the REORG utility if the user specified SHRLEVEL REFERENCE or CHANGE and DB2 estimates that the switch phase will not start by the deadline specified by the DEADLINE phrase. This message provides the user with an estimate of how far the REORG utility has proceeded in the indicated utility job step.

nnnnnn The current iteration number

www 'YES' or 'NO' according to whether write access is allowed in the current iteration of log processing

For the iteration before the previous iteration:

ha:ma:sa
The elapsed time for the iteration

nnnnna The number of log records processed in the iteration.

These values are 00:00:00 and 0 if the current iteration number is 1 or 2.

For the previous iteration:

hb:mb:sb
The corresponding information.

nnnnnb The corresponding information.

These values are 00:00:00 and 0 if the current iteration number is 1.

For the current iteration:

hc:mc:sc The estimated elapsed time

hd:md:sd
The elapsed time so far

nnnnnd The actual number of log records being processed

For the next iteration:

he:me:se The currently estimated elapsed time

nnnnne The currently estimated number of log records to be processed

Severity: 0 (informational)

System action: The message reflects the state of DB2 at the time of the -DISPLAY UTILITY command or the time when DB2 estimated that the switch phase would not start by the deadline specified by the DEADLINE phrase.

If this message was issued for -DISPLAY UTILITY, the system continues processing. A subsequent -DISPLAY UTILITY command reflects that processing.

DSNU384I *csect-name* **- MAXRO =** *aaaaaaaaaa* **SECONDS LONGLOG =** *llllllll* **DELAY =** *bbbbbbbbbb* **SECONDS**

Explanation: This message applies only to REORG with SHRLEVEL CHANGE. It supplements message DSNU105I, which can appear for any utility (including REORG).

This message is issued in response to a -DISPLAY UTILITY command.

It is also issued by the REORG utility if the user specified SHRLEVEL CHANGE and DB2 estimates that the switch phase will not start by the deadline specified by the DEADLINE phrase.

This message indicates the most recently specified values of certain parameters to REORG. For each such parameter, a previous invocation of the -ALTER UTILITY command might have specified a value. If no previous invocation of -ALTER UTILITY has specified a value, the most recently specified value is the value that was specified with the invocation of the REORG utility.

aaaaaaaaaa
A number of seconds, according to the most recently specified value of MAXRO. A value of "DEFER" denotes that the log file will be deferred indefinitely.

lllllll "CONTINUE," "TERM," or "DRAIN" according to the most recently specified value of LONGLOG

bbbbbbbbbb
A number of seconds, according to the most recently specified value of DELAY

Severity: 0 (informational)

System action: The message reflects the state of DB2 at the time of the -DISPLAY UTILITY command or the time when DB2 estimated that the switch phase would not start by the deadline specified by the DEADLINE phrase.

If this message was issued for -DISPLAY UTILITY, the system continues processing. If an -ALTER UTILITY command is then executed, a subsequent -DISPLAY UTILITY command reflects the results (if any) of the -ALTER UTILITY command.

DSNU385I *csect-name* **- LOG PHASE COMPLETE, ELAPSED TIME =** *hh/mm/ss*

Explanation: This message is issued at the normal completion of the LOG phase of the REORG utility. The time that has elapsed between the beginning and the completion of this phase was:
hh hours
mm minutes
ss seconds

Severity: 0 (informational)

System action: Processing continues to the next phase of the REORG utility.

DSNU386I *csect-name* **- LOG PHASE STATISTICS. NUMBER OF ITERATIONS =** *iiiiiiii*, **NUMBER OF LOG RECORDS =** *rrrrrrrr*

Explanation: This message is issued at the normal completion of the LOG phase of the REORG utility.

iiiiiiii The number of iterations of log processing that were performed. This number includes one or more iterations during which applications have read/write access plus one final iteration during which applications have read-only access.

rrrrrrrr The number of log records that were read.

Application of some of these log records might be unnecessary. Therefore, the LOG phase might apply only a subset of these log records to the shadow copy of the table space or partition.

Severity: 0 (informational)

System action: The LOG phase of the REORG utility is completed.

DSNU387I *csect-name* **- SWITCH PHASE COMPLETE, ELAPSED TIME =** *hh/mm/ss*

Explanation: This message is issued at the normal completion of the switch phase of the REORG utility. The time that has elapsed between the beginning and the completion of this phase was:
hh hours
mm minutes
ss seconds

Severity: 0 (informational)

System action: Processing continues to the next phase of the REORG utility.

DSNU388I *csect-name* **- BUILD2 PHASE COMPLETE, ELAPSED TIME =** *hh/mm/ss*

Explanation: This message is issued at the normal completion of the BUILD2 phase of the REORG utility. The time that has elapsed between the beginning and the completion of this phase was:
hh hours
mm minutes
ss seconds

Severity: 0 (informational)

System action: Processing continues to the next utility statement in the utility job step, if any.

DSNU389I *csect-name* **- BUILD2 PHASE STATISTICS. NUMBER OF NONPARTITIONED INDEXES =** *mmm*

Explanation: This message is issued at the normal completion of the BUILD2 phase of the REORG utility.

mmm The number of nonpartitioned indexes whose RIDs were corrected.

Severity: 0 (informational)

System action: The BUILD2 phase of the REORG utility is completed.

DSNU390I *csect-name* **INVALID CONVERSION FOR FIELD** *cccccccc*

Explanation: The input field type for column, 'cccccccc', is incompatible with its data type (as found in the catalog).

Severity: 8 (error)

System action: Utility processing is terminated.

User response: Correct the error, and resubmit the job. Refer to *DB2 Utility Guide and Reference* for compatible data types.

DSNU391I *csect-name* **- SORTBLD PHASE STATISTICS. NUMBER OF INDEXES =** *mmm*

Explanation: This message is issued at normal completion of the SORTBLD phase of the LOAD, REORG, or REBUILD INDEX utility. The number of indexes that were built is indicated as *mmm*.

Severity: 0 (informational)

System action: The SORTBLD phase of either the LOAD, REORG, or REBUILD INDEX utility is completed.

DSNU392I *csect-name* **- SORTBLD PHASE COMPLETE, ELAPSED TIME =** *hh:mm:ss*

Explanation: This message is issued at the normal completion of the SORTBLD phase of the LOAD, REORG, and REBUILD INDEX utility. The time that has elapsed between the beginning of the phase and the completion of this phase was *hh* hours, *mm* minutes, and *ss* seconds.

Severity: 0 (informational)

System action: Processing continues to the next phase of the LOAD, REORG, or REBUILD INDEX utility. If this is the last phase, processing continues to the next utility statement in the utility job step, if any.

DSNU393I *csect-name* **SORTBLD PHASE STATISTICS - NUMBER OF KEYS =** *nnnnnnnn* **FOR INDEX** *index-name* **PART** *part-num*

Explanation: This message is issued by the SORTBLD phase of the LOAD, REORG, or REBUILD INDEX utility to report the total number of index key entries used to build the indicated index partition.

Severity: 0 (informational)

System action: This message indicates the number of index key entries for the index partition being built. Both unique and nonunique key entries are counted. Rejected key entries are not counted.

User response: You can verify that the number reported is consistent with expectations.

DSNU394I *csect-name* **SORTBLD PHASE STATISTICS. NUMBER OF KEYS =** *nnnnnnnn* **FOR INDEX** *index-name*

Explanation: This message is issued by the SORTBLD phase of the LOAD, REORG, or REBUILD INDEX utility. The message is repeated for each index built by the LOAD, REORG, or REBUILD INDEX utility. It indicates the total number of index key entries used to build the indicated index. If the index is a partitioning index, the total will equal the sum of the number of index key entries reported in prior DSNU393I messages.

Severity: 0 (informational)

System action: This message indicates the number of index key entries for the index being built. Both unique and nonunique key entries are counted. Rejected key entries are not counted.

User response: You can verify that the number reported is consistent with expectations. For example, you can determine if it matches the number of records loaded for the table indexed by the indicated index, which will have been reported in a DSNU303I message issued during the RELOAD phase.

DSNU395I *csect-name* **INDEXES WILL BE BUILT IN PARALLEL, NUMBER OF TASKS =** *nnnn*

Explanation: This message is issued by the RELOAD phase of the LOAD or REORG or UNLOAD phase of REBUILD INDEX utility. The message indicates that the LOAD, REORG, or REBUILD INDEX utility is able to build indexes in parallel. The number of utility tasks used to build the indexes is indicated by *nnnn*.

Severity: 0 (informational)

System action: Processing of the RELOAD phase continues.

User response: You can verify that the number of tasks reported is consistent with expectations. For example, if 3 indexes are to be built, the user might expect to see number of tasks = 6.

DSNU396I *csect-name* **REPORT PHASE COMPLETE, ELAPSED TIME =** *hh/mm/ss*

Explanation: The REPORT phase of the LOAD utility has completed normally. The elapsed time for the execution was 'hh' hours, 'mm' minutes, and 'ss' seconds.

Severity: 0 (informational)

DSNU397I *csect-name* **NUMBER OF TASKS CONSTRAINED BY** *xxxxxxxx*

Explanation: This message indicates that while a utility is able to start subtasks to perform operations in parallel, the number of tasks it will use has been constrained by a lack of some resource. The constraining resource is specified by *xxxxxxxx*, which can have the following values and meanings:

VIRTUAL STORAGE
The utility could start additional tasks, but there is insufficient virtual storage in the address space in which the utility job is running to make it effective to do so.

VIRTUAL STORAGE BELOW
The utility could start additional tasks, but there is insufficient virtual storage below the line in the address space in which the utility job is running to make it effective to do so.

DD STATEMENTS
The utility could start additional tasks, but the user has specified DD statements for the data sets to be used by SORT for sort work data sets or for sort message data sets, and fewer sort work data set groups or sort message data sets have been specified than the maximum number of tasks the utility could otherwise start.

CONNECTIONS
The utility could start additional tasks, but there are not enough free threads available to DB2 to do so.

CPUS The utility could start additional tasks, but it would not be efficient to do so because it would overtax the available CPUs.

SWA=BELOW
The utility could start additional tasks, but
there is insufficient storage below the line in
the address space in which the job is running.

Severity: 0 (informational)

System action: Processing of the phase continues.

User response: No action is required. However, if you want to increase the number of tasks used by the utility in order to reduce the elapsed time of the utility job, then the possible actions to take depend on the value of *xxxxxxxx* in the message:

VIRTUAL STORAGE
Increase the amount of virtual storage available to the utility job by specifying a larger region size in the job JCL, altering the JES defaults for region size, or changing the system IEFUSI exit to allow a larger virtual region.

If the utility is using BSAM data sets for which the number of buffers have been specified (BUFNO specified in the DCB field of the DD statement for the data set), consider reducing the number of buffers.

VIRTUAL STORAGE BELOW
If possible, increase the amount of virtual storage below the line available to the utility job by specifying a larger region size in the job JCL or reducing the number of data sets allocated in the job step. It might not be possible to relieve this constraint.

DD STATEMENTS
Define additional groups of sort work data sets, or eliminate the SWnnWKmm DD statements and specify the SORTDEVT keyword (and, optionally, the SORTNUM keyword) to allow SORT to use dynamic allocation of its work data sets.

If the limitation is due to sort message data sets rather than sort work data sets, define additional sort message DD statements or eliminate the UTPRINnn DD statements and allocate UTPRINT to SYSOUT so that the sort message data sets can be dynamically allocated.

CONNECTIONS
Increase the number of threads available to DB2, or submit the job during a period of reduced DB2 activity.

CPUS Submit the job to a member that has more processors.

SWA=BELOW
Move the SWA above the 16 MB line for the
JOBCLASS. JES command, $DJOBCLASS(n),
will display whether or not SWA is set to
BELOW for CLASS n. The
$TJOBCLASS(n),SWA=ABOVE command will
set SWA to ABOVE the line. The $DSTCCLASS
and $TSTCCLASS commands can be used to
move SWA to ABOVE for the utilities that are
running in a WLM Stored Procedure
Environment. Please consult with the MVS
System Programmer on these changes.

DSNU398I *csect-name* **UNEXPECTED PROCESSING ERROR, REASON=X00E40318, ON TABLE** *ttttttt*

Explanation: When attempting to load a record into table 'ttttttt', an error was encountered. Refer to *DB2 Codes* for an explanation of reason code '00E40318'.

Severity: 8 (error)

System action: If the error was caused by a resource not being available, a DSNT500I message is issued.

The utility step is abended with a system X'04E' with an abend reason code of '00E40318'. If a DSNT500I message is not issued, an SVC dump is requested. A SYSABEND dump of batch storage is also requested.

The utility step is left in the stopped state.

User response: If a DSNT500I message was issued, refer to its description under that message number in this manual. If the unavailable resource becomes available, the utility step is restartable.

If a DSNT500I message is not issued, the error is an internal error. Notify the system programmer.

Operator response: An operator with SYSOPR authority can issue a TERM UTIL to free any held resources (this may not be necessary).

System programmer response: If you suspect an error

in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNU399I *csect-name* **- LOAD UTILITY ERROR SUMMARY REPORT**

Explanation: This message summarizes errors detected during the load job. A report is issued only if unique index or referential constraint violations were detected. The report contains one line for each error. The errors are listed first by error severity and within that by *idsrn*.

The report fields have the following meanings:

sev

The severity of the error reported on this line:

Severity	Type of error	Explanation
1	Primary	Primary referential integrity errors and all nonreferential integrity errors.
2	Secondary	Secondary referential integrity errors (that is, errors that result from a primary referential integrity error in an ancestor).

idsrn The relative record number in the input data set of the record causing the error reported on this line.

ddsrn The relative record number in the input data set of the record causing the error reported on this line, if discard processing is in effect. Otherwise, *ddsrn* is blank.

type A description of the type of error reported on this line, as follows:
- Conversion
- Field specification
- Validation
- Part Specification
- Unexpected
- Column specification
- Referential integrity
- Check constraint
- Duplicate key.

name1 The name of the table for which the record causing the error was targeted, if this information is pertinent for the type of error reported on this line. Otherwise, *name1* is blank. This field has nonblank values for the following error types:
- Conversion
- Validation
- Column specification
- Referential integrity (primary).
- Check constraint

name2 The name of the field or fanset involved in the error reported on this line, if this information is pertinent for the type of error reported. Otherwise, *name2* is blank. This field contains a field name for the following types of errors:
- Conversion
- Field specification
- Column specification.

For primary referential integrity errors, *name2* is a relationship name. For check constraint errors, *name2* is a check constraint name.

idsrn2 The input data set record number of a related error. For duplicate key errors, it gives the input data set record number of the record whose key is duplicated. For secondary referential integrity errors, it gives the input data set record number of the record whose primary referential integrity error cascaded to cause this secondary error. For other types of errors, this field is unused.

Severity: 4 (warning)

System action: Processing continues.

DSNU400I *csect-name* **COPY PROCESSED FOR** *obj-type obj-qual.obj-name* **NUMBER OF PAGES=***nnnnnn* **AVERAGE PERCENT FREE SPACE PER PAGE =** *ff.ff* **PERCENT OF CHANGED PAGES =** *pp.pp* **ELAPSED TIME=** *hh/mm/ss*

Explanation: This is an informational message to report the copy statistics. This message is issued by the COPY utility, and it can also be issued from LOAD or REORG if a full image copy has been requested during a table space load or reorganization. The statistics apply to pages included in the scope of the COPY request. This can either be a full table space, full index space (DSNUM(ALL)) or a partition (DSNUM(integer)). All the pages in the table space, index space, or partition can be copied (FULL(YES)), or only the changed pages can be copied (FULL(NO)). The message variables are:

ff.ff The average percentage of free space within the copied pages. This indicates the amount of change activity that can occur without significant impairment of reference performance.

nnnnnn The number of pages written out.

Image copies produced during LOAD or REORG will have more pages written than a full image copy produced by running the COPY utility on the same table space, because some pages will be written more than once.

pp.pp The percentage of changed pages. This indicates the amount of change activity that occurred since the last image copy or

reorganization. The percentage of changed pages is obtained by dividing the number of changed pages by the number of pages in the object. Preformatted pages consisting of all zeros are included in the count of pages scanned. Since these preformatted pages are not copied to the output data set, the number of pages scanned is usually larger than the number of pages copied.

When this message is issued for image copies produced during LOAD or REORG, this percentage will always be reported as 100%, because the utility is resetting and reformatting the entire table space.

System action: The COPY, LOAD or REORG utility continues processing this message is for information only. Message DSNU428I will be issued when the copy processing is complete.

User response: Reported statistics could suggest that either a RUNSTATS or REORG utility should be run on the table space.

DSNU401I *csect-name* **CONCURRENT COPY COMPLETE, ELAPSED TIME=** *hh*:*mm*:*ss*

Explanation: The DFSMS Concurrent Copy utility completed normally.

Severity: 0 (informational)

System action: Processing continues to the next utility statement in the utility job step. If there are no more utility statements, the utility job step is normally terminated, and the resources are released to their prior availability state. The utility identifier (UTILID) of the utility job step becomes available for reuse.

User response: No action is required.

DSNU402I *csect-name* **INCREMENTAL IMAGE COPY DISALLOWED FOR** *obj-type obj-qual.obj-name* **FULL IMAGE COPY WILL BE TAKEN**

Explanation: An incremental image copy for a table space or data set was requested. However, a full image copy is required. Possible reasons include:

- The request was for one of the DB2-maintained table spaces DSNDB01.DBD01, DSNDB06.SYSCOPY, or DSNDB01.SYSUTILX.
- A LOAD or REORG utility was run on the table space or data set with no subsequent full image copy.
- No full image copy was found for the requested sites.
- Copy pending status was set for the requested table space or data set.
- The request was for an entire table space, but the most recent image copy for the table space is on the data set level.
- The previous image copy is a DFSMS concurrent copy.

The request for an incremental image copy is ignored, and a full image copy is taken.

Severity: 4 (warning)

System action: The request to take an incremental image copy is treated as a request to take a full image copy.

User response: The capacity of the target image copy data set might not be sufficient if its size was established assuming an incremental image copy. If this condition occurs, terminate and then resubmit the job step.

DSNU403I *csect-name* **- FULL AND INCREMENTAL COPIES OF THE SAME TABLESPACE ON THE SAME TAPE DISALLOWED DDNAME** *ddname*

Explanation: The COPY utility has been invoked to take an incremental image copy of a TABLESPACE. The specified output data set is on a tape which contains another FULL or INCREMENTAL image copy of the same TABLESPACE. This is not allowed.

Severity: 8 (error)

System action: The utility will terminate with a return code of 8, and will not take the image copy or update the SYSCOPY table.

User response: Change the VOLSER for the image copy output data set or use the MODIFY RECOVERY utility to delete SYSIBM.SYSCOPY records which are no longer needed for the same TABLESPACE and tape volume.

Problem determination: Use the SQL SELECT command to look at the SYSIBM.SYSCOPY table. If the conflicting image copy data sets on the output tape are no longer needed, use MODIFY RECOVERY to eliminate the related SYSIBM.SYSCOPY records. If the data sets are still needed, you must place the new image copy on a different tape or you must take a FULL image copy.

DSNU404I *csect-name* **LOCAL SITE AND RECOVERY SITE INCREMENTAL IMAGE COPIES ARE NOT SYNCHRONIZED FOR** *obj-type obj-qual.obj-name*

Explanation: No incremental image copy can be made if

1. The incremental image copy is requested only for a site other than the current site.
2. The incremental image copies are requested for both sites but the most recent full image copy was made only for the current site.

3. The incremental image copies are requested for both sites and the most recent full image copies were made for both sites, but between the most recent full image copy and current request, all incremental image copies were made for current site only.

Severity: 8(error)

System action: The utility will terminate with a return code of 8, and will not take the image copy or update the SYSCOPY table.

User response: Take another full image copy of the TABLESPACE for both LOCAL site and RECOVERY site or change the request to make incremental image copy only for current site.

DSNU406I **FULL IMAGE COPY SHOULD BE TAKEN FOR BOTH LOCAL SITE AND RECOVERY SITE FOR** *obj-type obj-qual.obj-name*

Explanation: The requested full image is for one site only but history shows that there were image copies taken for both local site and recovery site and the table space is in copy pending status, or point in time recover has been done on the table space, or the table space is in copy pending status and the requested full image copy is for a site other than the current site. Be aware of the following situations:

1. If the requested copy is made for current site, then the other site can only do point in time recovery.
2. If the requested copy is for the site other than the current site, then the TABLESPACE will be in the same state as before COPY has been made.

Severity: 4 (warning)

System action: COPY, LOAD, or REORG utility will continue to process the image copy.

User response: Take another full image copy of the TABLESPACE for both LOCAL site and RECOVERY site to guarantee the correct recovery on both sites or check DSNZPARM to ensure that the current site is the site at which the image copy is being taken.

DSNU407I *csect-name* **- DATA SET ALREADY USED FOR PREVIOUS IMAGE COPY DDNAME** *ddname*

Explanation: A COPY, LOAD, or REORG utility has been invoked to take an image copy to a data set which is a duplicate of one already recorded in SYSCOPY. If the specified data set is cataloged, a matching DSNAME exists. If the specified data set is not cataloged, a matching DSNAME, DSVOLSER, and FILESEQNO exists.

Severity: 8 (error)

System action: The utility will terminate with a return code of 8, and will not take the image copy or update the SYSCOPY table.

User response: Change the name of the data set or use MODIFY utility to delete the old SYSCOPY record containing the duplicate data set.

DSNU408I *csect-name* **-** *dsn* **EXISTS WITH -VOLUME=***vol***, FILE SEQUENCE-NUMBER=***fseq***, RBA=***rba*|*lrsn***, DATE=***date***, TIME=***time*

Explanation: If this message is issued after DSNU403I, it provides information from the SYSCOPY catalog table about the previously created image copy of the table space. If this message is issued after DSNU407I, it provides information from the SYSCOPY table about the duplicate data set that was detected during image copy processing.

For copies to DASD or MSS, *fseq* is 0; for copies of cataloged data sets, *vol* is (CATG).

Severity: 8 (error)

System action: The utility terminates with return code 8 and does not take the image copy or update the SYSCOPY table.

User response: Use the information provided to change the name of the data set or to locate the duplicate SYSCOPY record for deletion with the MODIFY RECOVERY utility.

DSNU409I *csect-name* **NO** *hardware-or-software* **SUPPORT FOR** *obj-type obj-qual.obj-name*

Explanation: This message is issued by the COPY or RECOVER utility for one of the following reasons.

- The COPY utility was invoked to take a DFSMS concurrent copy of a table space or index space that resides on a device with no hardware or software support for the DFSMSdss DUMP command with the CONCURRENT option.
- The RECOVER utility invoked the DFSMSdss RESTORE command without the supporting software.

Severity: 4 (for RECOVER) or 8 (for COPY)

System action: The RECOVER utility falls back to the previous image copy data set and continues. The COPY utility job terminates.

User response: If a list of objects is specified in the COPY job, remove the object that has no hardware or software support from the list. Resubmit the job or use the DB2 COPY utility instead of the DFSMS Concurrent Copy utility.

DSNU410I *csect-name obj-type obj-qual.obj-name* **CONTAINED NO CHANGED PAGES**

Explanation: An incremental image copy (COPY option FULL(NO)), which copies only changed pages,

was requested. However, no changed pages were found.

Severity: 4 (warning)

This message reflects a warning condition that did not prevent subsequent processing.

System action: This message indicates normal completion of the image copy utility. Processing continues to the next utility statement in the utility job step, if any. If there are no more utility statements, the utility job step is normally terminated, and the resources are released to their prior availability state. The UTILID of the utility job step becomes available for reuse.

Because no output was produced, the SYSIBM.SYSCOPY catalog table remains unchanged.

The requester may want to determine why the apparent assumption that a change occurred was not true.

Operator response: The target copy data set identified by the DD statement nominated by the COPY COPYDDN option was not required. It was not tracked in the DB2 catalog table SYSIBM.SYSCOPY. It need not be kept for subsequent re-call.

DSNU411I *csect-name* **UNABLE TO LOGICALLY COMPLETE** *obj-type obj-qual.obj-name* **RETURN CODE** *return-code* **REASON CODE** *reason-code*

Explanation: The DFSMSdss DUMP command with the CONCURRENT option encountered an error during the initialization of the target object being copied. The return code and reason code are from the DFSMSdss DUMP command.

Severity: 8

System action: The utility job terminates.

User response: See the return code and reason code in *z/OS DFSMSdss Storage Administration Reference* to determine the problem. Resubmit the job after resolving the problem.

DSNU412I *csect-name* **UNABLE TO PHYSICALLY COMPLETE** *obj-type obj-qual.obj-name* **RETURN CODE** *return-code*

Explanation: The DFSMSdss DUMP command with the CONCURRENT option encountered an error during the backup process. The return code is from the DFSMSdss DUMP command.

Severity: 8

System action: The utility job terminates.

User response: See the return code description in *z/OS DFSMSdss Storage Administration Reference* to determine the problem. Resubmit the job after resolving the problem, or use the DB2 Copy utility instead of the DFSMS Concurrent Copy utility for the failed object.

DSNU413I *csect-name* **- CONCURRENT COPY SUCCESSFUL FOR** *obj-type obj-qual.obj-name*

Explanation: The DFSMS Concurrent Copy utility completed normally for the named object. This message is repeated for every copied object in the list.

Severity: 0 (informational)

System action: Processing continues to the next object in the list. If there are no more objects, the utility ends normally with message DSNU401I.

User response: No action is required.

DSNU414I *csect-name* **UNABLE TO RETRIEVE RESTART/REDO LRSN** *obj-type obj-qual.obj-name*

Explanation: This message is issued by the DB2 COPY utility when an error is encountered during the retrieval of a log record sequence number (LRSN).

This error can occur in these situations:

- Another DB2 in a data sharing group terminated abnormally while trying to update the table space or index space.
- A failure occurred while trying to communicate with other DB2s in the data sharing group.

Severity: 8

System action: The utility job terminates.

User response: Issue a -DISPLAY DATABASE command with the USE option to determine if any other DB2 systems hold a retained lock on the table space or index space (status = RSTRT). If so, you must restart that DB2 system before the COPY job can be successfully run.

DSNU415I *csect-name* **INVALID OUTPUT BLOCK SIZE SPECIFIED DDNAME** *ddname*

Explanation: Valid block sizes for an image copy or mergecopy data set are multiples of 4KB and less than the device limitation.

Severity: 8 (error)

System action: Utility processing is terminated.

User response: Correct the JCL statement, or use the default. Resubmit the request in a new utility job step. Refer to *DB2 Utility Guide and Reference* for a discussion of image copy and work data set constraints.

DSNU416I *csect-name* **THE OUTPUT IMAGE COPY DATA SET CANNOT BE PARTITIONED. DDNAME** *ddname*

Explanation: The use of partitioned image copy data sets is restricted. The COPY, MERGECOPY, LOAD, or REORGutility job that was submitted has a partitioned output image copy data set, specified by the COPYDDN ddname or RECOVERYDDN ddname parameter.

Severity: 8 (error)

System action: Utility processing is terminated. The image copy will not be taken or the merge will not be done, and the SYSIBM.SYSCOPY table will not be updated.

User response: Correct the utility job by specifying a nonpartitioned data set as the output image copy data set.

DSNU417I *csect-name* **THE NUMBER OF VOLUMES USED BY THE IMAGE COPY DATA SET** *data-set-name* **HAS REACHED THE LIMIT**

Explanation: The image copy data set reached the limit of 255 volumes and another volume is required.

Severity: 8 (error)

System action: The utility job is terminated.

User response: You might want to use the DSNUM option of the COPY utility to identify a partition or data set within the table space to be copied, or specify a partition number to be loaded or reorganized if the copy is being produced from a LOAD or REORG.Refer to *DB2 Utility Guide and Reference* for a description of the COPY utility and the DSNUM option, and descriptions of the LOAD and REORG utilities and the PART option.

DSNU418I *csect-name* **CANNOT TAKE IMAGE COPY AT DATASET LEVEL FOR NONPARTITIONED TABLESPACE** *obj-qual.obj-name* **IN COPY PENDING STATUS**

Explanation: An image copy at the data set level was requested. However, it is not allowed for one of the following reasons:

- The table space is in copy pending status.
- The table space is not suitable for image copy at the data set level.

A copy of the entire table space is required.

Severity: 8 (error)

System action: The COPY utility completes unsuccessfully with return code 8.

System programmer response: You must take a full image copy of the entire table space by specifying the DSNUM ALL option with the COPY utility.

DSNU420I *csect-name* **INCORRECT SHARE LEVEL SPECIFIED FOR TABLE SPACE** *database-name.tablespace-name* **DSNUM** *dsnum*

Explanation: When copying a table space that is not recoverable, the share level (SHRLEVEL) must be REFERENCE.

If the object is an entire table space, 'DSNUM *dsnum*' is not displayed.

Severity: 8 (error)

System action: Utility processing is terminated.

User response: Correct the SHRLEVEL to REFERENCE, and resubmit the COPY utility in a new utility job step. Refer to *DB2 Utility Guide and Reference* for a discussion of COPY utility constraints.

DSNU421I *csect-name* **- START OF DFSMS MESSAGES**

Explanation: This message indicates the beginning of the DFSMS messages; DSNU422I indicates the end.

System action: No action is required, but subsequent DFSMS messages might require action. See the DFSMS documentation for furhter information.

DSNU422I *csect-name* **- END OF DFSMS MESSAGE**

Explanation: This message indicates the end of the DFSMS messages; DSNU421I indicates the beginning.

System action: No action is required, but preceding DFSMS messages might require action. See the DFSMS documentation for furhter information.

DSNU423I *csect name* **CANNOT TAKE CONCURRENT COPY WITH SHRLEVEL CHANGE OPTION FOR** *obj-type obj-qual.obj-name* **WITH GREATER THAN 4KB PAGE SIZE**

Explanation: The DFSMS Concurrent Copy utility is serialized at the extent level. The greater than 4KB page size condition might span more than one extent. This might cause the output data set to contain inconsistent data.

Severity: 8 (error)

System action: The COPY utility completes unsuccessfully with return code 8.

System programmer response: You must take a concurrent copy with the SHRLEVEL REFERENCE option or a DB2 image copy if you prefer the SHRLEVEL CHANGE option.

Convert the data set attribute Control Interval size of the associated tablespace from the 4K Control Interval into the one that matches the page size. DB2 Utilities REORG Tablespace, LOAD with REPLACE, and RECOVER can be used to convert the table space.

DSNU424I *csect-name* **INCOMPATIBLE MAINTENANCE DETECTED. REFERENCE APAR NUMBER** *apar* **DO THE FOLLOWING:**

Explanation: This message indicates that there is a maintenance incompatibility detected across restart. This message is followed by the actions the user should take to resolve the incompatibility.

apar The number of the APAR with closing text that describes the incompatibility.

System action: Utility execution is abended.

User response: Refer to the action text in the message.

System programmer response: Refer to the APAR closing text.

DSNU425I *csect-name obj-type obj-qual.obj-name* **DOES NOT HAVE THE COPY YES ATTRIBUTE**

Explanation: The utility cannot process the identified index or index space because it was not defined with the COPY YES attribute. Without the COPY YES attribute, the index cannot be specified as a target for the RECOVER, COPY, or REPORT utilities. If the index needs to be rebuilt from the data in the table space, use the REBUILD INDEX utility instead.

Severity: 4 (warning) or 8 (error)

System action: The RECOVER utility terminates processing with a return code of 8. The COPY utility skips this object but continues processing the other objects in the list. The final COPY utility return code will be 8. The REPORT utility issues this message with a return code of 4 and continues processing other objects (if any).

User response: Determine whether the index should be altered to have the COPY YES attribute. If so, alter the index then rerun the utility job. Otherwise remove the index from the utility statement.

DSNU426I *csect-name* **CANNOT COPY ONE PIECE OF NON-PARTITIONING** *obj-type obj-qual.obj-name*

Explanation: The copy of a nonpartitioning index at the data set level is not allowed.

Severity: 8 (error)

System action: The COPY utility skips this object but continues processing the other objects in the list. The final COPY utility return code will be 8.

User response: Either remove the DSNUM option so that it defaults to DSNUM(ALL) or modify your DSNUM option to DSNUM(ALL).

DSNU427I *csect-name* **OBJECTS WILL BE PROCESSED IN PARALLEL, NUMBER OF OBJECTS =** *nnnn*

Explanation: This message is issued by COPY or RECOVER utilities. The message indicates that COPY or RECOVER is able to process the objects in the list in parallel. The number of objects used is indicated by *nnnn*.

Severity: 0 (informational)

System action: Processing of the utility continues.

User response: You can verify that the number of objects processed is consistent with expectations.

DSNU428I *csect-name* **DB2 IMAGE COPY SUCCESSFUL FOR** *obj-type obj-qual.obj-name*

Explanation: The image copy of the object processed by the COPY, LOAD, or REORG utility has completed successfully and the image copy data set information has been recorded in the SYSIBM.SYSCOPY catalog table.

System action: The COPY utility continues its processing on the next object in the list. The LOAD and REORG utilities continue processing in the subsequent phases.

DSNU429I *csect-name* **RECOVERY OF** *objecttype database-name.tablespace-name* **MUST BE EXECUTED AT THE PARTITION LEVEL**

Explanation: You must recover the partitioned object at the partition level. You may receive this message if you are using COPY with the CONCURRENT option. The use of online REORG (SHRLEVEL REFERENCE or CHANGE) with the FASTSWITCH YES option has created a mixture of 'I0001' and 'J0001' instance nodes (fifth qualifier) for the underlying LDS's for the partitions of this object.

Severity: 0 (informational)

System action: Utility processing continues.

DSNU430I *csect-name* **- ERROR ALLOCATING TEMPORARY FILTER FOR** *obj-type obj-qual.obj-name* **DSN=***name* **CODE=X'***reason-code***'**

Explanation: SVC 99 returned an error during the dynamic allocation of a temporary filter data set. This message is followed by text that explains the reason-code that is shown.

The COPY utility was specified with the CONCURRENT option, and has detected that the number of data sets to be processed will exceed the DFSMSdss DUMP command limitation of 255. The utility will dynamically allocate a temporary filter data set for the object and generate a DFSMSdss DUMP command with the FILT option.

Severity: 8 (error)

System action: Utility processing halts for this object.

The utility job completes with a return code of 8.

If the utility job step includes an OPTIONS control statement with the EVENT(ITEMERROR,SKIP) option, the utility will process the next item in the list.

User response: Correct the error that is preventing dynamic allocation, and resubmit the utility job. In some cases where dynamic allocation may not be possible, consider using the FILTERDDN option on the COPY utility statement.

DSNU431I *csect-name* **- ERROR ALLOCATING TEMPORARY FILTER FOR** *obj-type obj-qual.obj-name* **DSNUM=***name* **CODE=X'***reason-code***'**

Explanation: An error returned during the dynamic allocation of a temporary filter data set. This message is followed by text that explains *reason-code*.

The RECOVER utility was run with the CURRENTCOPYONLY option and has detected that the number of data sets to be processed exceeds the DFSMSdss RESTORE command limitation of 255. DB2 dynamically allocates a temporary filter data set for the object and generates a DFSMSdss RESTORE command with the FILT option.

Severity: 8 (error)

System action: If the utility job step includes an OPTIONS control statement with the EVENT(ITEMERROR,SKIP) option, the utility processes the next item in the list. Otherwise, utility processing stops with return code of 8.

User response: Correct the error that prevents dynamic allocation and resubmit the utility job. If dynamic allocation is not possible, remove be possible, remove the CURRENTCOPYONLY option on the RECOVER utility statement.

DSNU432I *csect-name* **- ERROR DURING DFDSS RESTORE OBJECT FOR** *obj-type obj-qual.obj-name***. THE OBJECT IS NOT RECOVERED**

Explanation: The RECOVER utility received an error notification from DFdss on object *obj-qual.obj-name*. The object has not been restored, but recovery of the other objects continues.

Severity: 4 (warning)

System action: RECOVER processes the other objects.

When RECOVER completes, it displays all objects for which CURRENTCOPYONLY was specified.

User response: Remove the CURRENTCOPYONLY keyword from the utility control statements.

DSNU434I *csect-name* **- TEMPLATE** *template-name* **HAS THE STACK YES OPTION ON** *dd-name* **DATA SET WHICH IS NOT SUPPORTED FOR** *utility-name***.**

Explanation: Template processing has detected tape stacking on an output data set which is not supported.

Severity: 8 (error)

System action: If the utility job step includes an OPTIONS control statement with the EVENT(ITEMERROR,SKIP) option, the utility processes the next item in the list. The utility will end with RC8 and subsequent utility statements in the job step are not executed. Otherwise, if EVENT(ITEMERROR,HALT) is specified or defaulted to, the utility processing stops immediately with a return code of 8 and subsequent utility statements in the job step are not executed.

User response: Remove the STACK YES option from the TEMPLATE statement, then resubmit the job.

DSNU440I *csect-name* **IMAGE COPY INFORMATION**

Explanation: This message gives statistical information about the results of running the COPY utility. It applies to a table space, partition, or data set, and is based on the information in the space page map at the time the utility is executed. The output contains the following information:

DBNAME
The name of the database to which the table space belongs.

TSNAME
The table space specified in the utility statement.

DSNUM
The partition or data set number of the table space. The value is ALL if information is about the entire table space.

***nn*KB PAGES**
The total number of *nn*KB pages in the partition, data set, or entire table space. This value includes preformatted pages that contain all zeroes.

EMPTY PAGES
The total number of empty pages in the data set of the segmented table space, or entire

segmented table space. Empty pages can result from deletes, or are pages that are not yet formatted. They are not copied by the utility.

| **CHANGED PAGES**
The number of pages in the partition, data set, or table space that were changed since the last time an image copy was created. If an incremental image copy is taken, this value is the number of pages that are copied.

| **PERCENT OF CHANGED PAGES**
The percent of changed pages in the table space, data set, or partition specified.

ICTYPE
The type of image copy recommended for the partition, data set, or table space specified. ICTYPE is based on the comparison between the percent of changed pages and the change limit values. ICTYPE values:

I If the percent of changed pages is greater or equal to the low CHANGELIMIT value and less than the high CHANGELIMIT value. An incremental image copy is recommended, or COPY takes one if the REPORTONLY option is not specified.

F If the percent of changed pages is equal to or greater than the high CHANGELIMIT value. A full image copy is recommended, and COPY takes one if the REPORTONLY option is not specified.

NONE If no pages were changed. An image copy is not recommended, and COPY does not take one.

blank For the partition or data set if DSNUM ALL is specified, or if CONCURRENT is specified and the percent of changed pages is less than the CHANGELIMIT value.

Severity: 0 (informational)

System action: Utility processing continues

User response: If the message indicates, "INFORMATION CANNOT BE DETERMINED" for a table space, partition, or data set, examine the utility output for the DSNU441I message which lists the page numbers of the broken pages. See message DSNU441I for further instructions on repairing pages.

DSNU441I **csect-name BROKEN PAGE FOUND**
DBNAME = dbname TSNAME =
tsname DSNUM = dsnum PAGE = page
ERRORCODE = *error-code*

Explanation: The COPY utility found a broken space map page.

dbname
The name of the database to which the table space belongs.

tsname The table space containing the broken page.

dsnum The partition or data set number of the table space containing the broken page.

page The hexadecimal page number of the broken page.

error-code
The specific cause of the problem. Only some
CSECT modules issue an error code. If the
CSECT module does not issue an error code,
the message that you receive will not list the
ERRORCODE keyword and value.

Severity: 8 (error)

System action: Utility execution continues.

User response: The data sets identified are in an indeterminate state. Use RECOVER, RECOVER PAGE, or the REPAIR utility to correct the broken page. Refer to Part 2 of *DB2 Diagnosis Guide and Reference*.

DSNU442I *csect-name* **CHANGELIMIT REPORT INFORMATION UNAVAILABLE FOR** *obj-type obj-qual.obj-name* **WITH TRACKMOD NO**

Explanation: This message is issued in response to a COPY utility. The CHANGELIMIT option cannot produce statistical information for a table space or a partition that is defined with TRACKMOD NO, or before a copy has been taken after the TRACKMOD attribute has been changed from NO to YES.

Severity: 4 (informational)

System action: If REPORTONLY is not specified then a full image copy will be taken.

User response: To obtain complete statistical information run CHANGELIMIT REPORTONLY after a copy has been taken.

DSNU443I **csect-name type CHANGE LIMIT MET FOR** *obj-type obj-qual.obj-name*

Explanation: The COPY utility is indicating whether the percent of changed pages in the specified table space, partition, or data set meets the CHANGELIMIT values. *type* values:

NO If no CHANGELIMIT met.

INCREMENTAL
If the percent of changed pages is greater than the low CHANGELIMIT value and less than the high CHANGELIMIT value.

FULL If the percent of changed pages is greater than or equal to the the high CHANGELIMIT value.

Severity: The return codes are:

1 (informational)
If no CHANGELIMIT was met.

2 (informational)
If the percent of changed pages is greater than the low CHANGELIMIT and less than the high CHANGELIMIT value.

3 (informational)
If the percent of changed pages is greater than or equal to the high CHANGELIMIT value.

System action: Utility processing continues.

User response: If the REPORTONLY option is specified and the result of this job step is not used to conditionally execute other job steps, examine the utility messages for this table space, partition, or data set to determine if it has changed to such a degree that an image copy or other step should be performed.

DSNU445I **csect-name** *ictype* **IMAGE COPY WILL BE TAKEN FOR** *obj-type obj-qual.obj-name*

Explanation: The COPY utility is indicating whether the percent of changed pages in the table space meets the change limit value, and whether an image copy will be taken. *ictype* values:

NO If no CHANGELIMIT value met.

INCREMENTAL
If the percent of changed pages is greater than the low CHANGELIMIT value and less than the high CHANGELIMIT value.

FULL If the percent of changed pages is greater than or equal to the high CHANGELIMIT value.

Severity: 0 (informational)

System action: Utility processing continues. An image copy is taken if a change limit is met. If the low CHANGELIMIT value is met, then an incremental copy is taken. If the high CHANGELIMIT value is met, then a full image copy is taken.

DSNU446I **csect-name - CHANGELIMIT REPORT INFORMATION UNAVAILABLE FOR** *obj-type obj-qual.obj-name*, **A FULL IMAGE COPY MUST BE TAKEN**

Explanation: The changelimit dataset statistical information was requested, but is unavailable. A full image copy is required. Possible reasons are as follows:

- The request was for one of the DB2-maintained table spaces: DSNDB01.DBD01, DSNDB06.SYSCOPY, or DSNDB01.SYSUTILX.
- A LOAD or REORG utility was run on the table space or data set with no subsequent full image copy.
- No full image copy was found for the requested sites.
- Copy pending status was set for the requested table space or data set.
- The request was for an entire table space, but the most recent image copy for the table space is on the data set level.
- The previous image copy is a DFSMS concurrent copy.

The statistical information report will not be generated. If REPORTONLY was omitted, then a FULL image copy is taken.

Severity: 4 (warning)

System action: The request for CHANGELIMIT without REPORTONLY is treated as a request to take a full image copy. The request for CHANGELIMIT with REPORTONLY will not generate the report.

User response: If REPORTONLY is specified, then submit a job to take a full image copy.

DSNU448I *csect-name* **- SCOPE PENDING WAS**
SPECIFIED, BUT NO OBJECT WITH
PENDING STATUS WAS FOUND

Explanation: SCOPE PENDING was specified for the
utility job, however no objects existed in the list with a
pending status.

Severity: 0 (informational)

System action: Utility processing continues.

DSNU454I *csect-name* **COPY MERGE COMPLETE. NUMBER OF COPIES =** *iiiii*, **NUMBER OF COPIES MERGED =** *cccccc*, **TOTAL NUMBER OF PAGES MERGED =** *ttttt*, **ELAPSED TIME =** *hh/mm/ss*

Explanation: This message marks the normal completion of the MERGECOPY utility. The variables are:

iiiii The number of copies found.

cccccc The number of image copies that were merged. The number merged can be less than the number found if a work data set was not specified using the MERGECOPY WORKDDN option.

ttttt The total number of pages that were merged.

The merged incremental copies were consolidated either into a single new incremental image copy or into a new full image copy, depending upon the MERGECOPY NEWCOPY option. The DB2 catalog table SYSIBM.SYSCOPY was updated to reflect the results.

This message may indicate that prior data set entries were deleted and, consequently, that backup data set resources were freed.

Severity: 0/4 (informational/warning)

System action: Processing continues to the next utility statement in the utility job step, if any. If there are no more utility statements, the utility job step is normally terminated, and the resources are released to their prior availability state. The UTILID of the utility job step becomes available for reuse.

User response: You can review the reported statistics to assess the effectiveness of performing incremental image copies. You can examine the SYSIBM.SYSCOPY catalog table to review the incremental and full image copies that are subject to DB2 recovery.

The full or incremental image copy data set identified by the DD statement specified in the COPYDDN option was recorded in SYSIBM.SYSCOPY. DB2 assumes that the image copy data set is now available for subsequent recovery. Ensure that the data set is available for recovery.

DSNU458I *csect-name* **NO IMAGE COPIES TO MERGE**

Explanation: This message is issued in response to a MERGECOPY request. The MERGECOPY utility was invoked to merge existing incremental image copies. However, in checking the DB2 catalog table SYSIBM.SYSCOPY, no incremental image copies were found.

Severity: 4 (warning)

A severity 4 condition is given to alert the invoker that processing did not proceed as defined. However, processing is proceeding; nothing was encountered that would invalidate further processing.

System action: This message indicates the completion of the MERGECOPY utility. Processing continues to the next utility statement in the utility job step, if any. If there are no more utility statements, the utility job step is normally terminated, and the resources that were held become available for REFERENCE processing. The UTILID of the utility job step becomes available for reuse.

The requester may want to determine why the assumption that incremental copies existed was not true.

Operator response: The target copy data set identified by the DD statement nominated by the MERGECOPY COPYDDN option was not required. It was not tracked in the DB2 catalog table SYSIBM.SYSCOPY. It need not be kept for subsequent re-call.

DSNU459I **SYSCOPY P RECORD ENCOUNTERED FOR** *obj-type obj-qual.obj-name*, **PIT RBA = xxxxxx**

Explanation: The utility encountered a SYSCOPY P record which requires that the SYSCOPY record at the given RBA or LRSN be used. The utility properly traverses the SYSCOPY records following this SYSCOPY P record.

Severity: 0 (informational)

User response: For further information on how the SYSCOPY records are traversed, see Part 4 (Volume 1) of *DB2 Administration Guide*.

DSNU460I *csect-name* **IMAGE COPIES INCONSISTENT. MERGECOPY REQUEST REJECTED**

Explanation: MERGECOPY was requested for an entire table space. However, the most recent image copy for the associated table space was on the data set level. In addition, if the associated table space was partitioned, this might indicate that any or all of its partitions were being reorganized or loaded without logging after the MERGECOPY starting point. The MERGECOPY starting point is the most recent full image copy for the entire table space.

Severity: 4 (warning)

A severity 4 condition is given to alert the user that processing did not proceed as defined. However, processing is proceeding; nothing was encountered that would invalidate further processing.

System action: This message indicates the completion of the MERGECOPY utility. Processing continues to the next utility statement in the utility job step, if any. If there are no more utility statements, the utility job step is normally terminated, and the resources that were held become available for REFERENCE processing. The UTILID of the utility job step becomes available for reuse.

No output was produced and the SYSIBM.SYSCOPY catalog table remains unchanged.

User response: A full image copy of the associated table space should be taken. However, if it is preferred that a recovery be done on the data set level, a MERGECOPY job should be submitted by specifying 'DSNUM(data set number)' for each data set for which MERGECOPY is required.

DSNU461I *csect-name* **PRIMARY FULL COPY DATA SET NOT FOUND, BACKUP WILL BE USED INSTEAD**

Explanation: The primary full image copy cannot be found. The backup full image copy will be located and used for the merge process.

Severity: 0 (informational)

User response: If the MERGECOPY utility can use the backup full image copy, the utility continues to process. If no backup full image copy can be used, the utility terminates.

DSNU462I *csect-name* **MERGECOPY DISALLOWED FOR REQUESTED TABLE SPACE**

Explanation: MERGECOPY was requested for table space DSNDB01.DB01, DSNDB01.SYSUTIL, or DSNDB06.SYSCOPY. MERGECOPY is not permitted for one of the following reasons:

- Only full image copies are allowed on these DB2-maintained tables.
- There is no record of the full image copies in the SYSIBM.SYSCOPY catalog; the copy information is in the DB2 log.

Severity: 8 (error)

System action: Further utility processing is terminated.

System programmer response: Refer to Part 4 (Volume 1) of *DB2 Administration Guide* for more information about MERGECOPY utility constraints.

DSNU463I *csect* **- THE** *xxxxxxx* **IMAGE COPY DATA SET** *data-set-name* **WITH DATE=***dddddd* **AND TIME=***tttttt***, IS PARTICIPATING IN MERGECOPY.**

Explanation: This informational message indicates the image copy that is used in the merge process. The variables have the following meanings:

xxxxxxxx
PRIMARY or BACKUP, indicating a primary or backup image copy

data-set-name
The image copy data set name

dddddd
The image copy date

tttttt The image copy time

Severity: 0 (informational)

User response: The utility continues to process normally.

DSNU464I **THIS IS A** *sitetype; outputtype* **WITH NEWCOPY NO IS NOT ALLOWED.**

Explanation: The 'NEWCOPY NO' option to create incremental image copies is not supported for the current site. This error can be caused by:

- Running the MERGECOPY utility at *sitetype* LOCALSITE, but selecting the RECOVERYDDN with the 'NEWCOPY NO' option.
- Running the MERGECOPY utility at *sitetype* RECOVERYSITE, but selecting the COPYDDN with the 'NEWCOPY NO' option.

Severity: 8 (error)

System action: The command cannot be executed.

User response: Correct the error and resubmit the job. Refer to *DB2 Utility Guide and Reference* for detailed information.

DSNU470I **TABLESPACE DSNDB01.SYSUTILX, ITS TABLES AND INDEXES CANNOT BE INCLUDED IN A LIST OF OBJECTS.**

Explanation: This message is issued by the running utility when a control statement contains a list of objects, or references a LISTDEF, which includes any of the following SYSUTILX related objects: TABLESPACE DSNDB01.SYSUTILX, INDEXSPACE DSNDB01.DSNLUX01, INDEXSPACE DSNDB01.DSNLUX02, TABLE SYSADM.SYSUTIL, TABLE SYSADM.SYSUTILX, INDEX SYSADM.DSNLUX01 or INDEX SYSADM.DSNLUX02.

Severity: 8(error)

System action: The utility job terminates.

User response: Remove the invalid object from the control statement and run the utility against DSNDB01.SYSUTILX, its tables and indexes separately.

DSNU471I *csect-name* **- COPY PENDING ON TABLESPACE** *databasename.tablespacename* **PROHIBITS PROCESSING**

Explanation: An attempt was made to execute a utility against a table space. Either the table space was in copy pending state or DSNDB06.SYSCOPY was in copy pending state.

Severity: 8 (error)

System action: Utility processing is terminated.

User response: Remove the table space from the copy pending state and resubmit the job.

Problem determination: Refer to Part 4 (Volume 1) of *DB2 Administration Guide* for a discussion of the database recovery function.

DSNU472I *csect-name* **- TABLESPACE** *tablespace-name* **HAS WRITE ERROR RANGE OR IS IN DEFERRED RESTART.**

Explanation: The attempt to externalize table space 'tablespace-name' pages to the media has failed.

Severity: 4 (warning)

System action: QUIESCE still continues but you cannot make an image copy of the table space. Processing continues for all indexes within the table space and continues to the next table space in the tablespace-list, if any.

User response: Run DISPLAY DATABASE to determine what exception state, if any, would prevent image copy on the table space.

Problem determination: Refer to Part 4 (Volume 1) of *DB2 Administration Guide* for a discussion of the database recovery function.

DSNU473I *csect-name* **- UNABLE TO EXTERNALIZE TABLESPACE** *tablespace-name* **PAGES**

Explanation: The attempt to externalize TABLESPACE 'tablespace-name' pages to the media has failed.

Severity: 4 (warning)

System action: QUIESCE still continues but the table space cannot be volume-dumped.

Problem determination: The problem may be an I/O error.

DSNU474I *csect-name* **- QUIESCE AT RBA** *rrrrrrrrrrr* **AND AT LRSN** *xxxxxxxxxx*

Explanation: This message displays the log RBA and LRSN at the time of the quiesce. This value is unique for all quiesced table spaces in the list and is recorded in the START_RBA column of the SYSCOPY catalog table entry for each quiesced table space in the list.

Severity: 0 (informational)

System action: Processing continues.

DSNU475I *csect-name* **- QUIESCE UTILITY COMPLETE, ELAPSED TIME=***tttttt*

Explanation: The message marks the normal completion of the QUIESCE utility. The elapsed clock time that the QUIESCE utility uses is shown by 'tttttt'.

Severity: 0 (informational)

System action: Processing continues to the next utility statement in the utility job step, if any. If there are no more utility statements, the utility job step is normally terminated, and the resources are released to their prior available state. The UTILID of the utility job step becomes available for reuse.

DSNU476I *csect-name* **- INDEX** *index-name* **HAS WRITE ERROR RANGE OR IS IN DEFERRED RESTART**

Explanation: The attempt to externalize index 'index-name' pages to the media failed.

Severity: 4 (warning)

System action: QUIESCE processing continues.

User response: Issue the DISPLAY DATABASE command to determine the exception state,

Problem determination: Refer to Part 4 (Volume 1) of *DB2 Administration Guide* for a discussion of the database recovery function.

DSNU477I *csect-name* **- QUIESCE SUCCESSFUL FOR** *obj-type obj-qual.obj-name*

Explanation: The QUIESCE for this object was successful. If the object is a table space, then the quiesce for all of its indexes was successful. This message will also be issued for each index with the COPY YES attribute over the quiesced table space. This message will be repeated for every quiesced table space in list.

Severity: 0 (informational)

System action: All data held in the DB2 buffer pool(s) for the table space and its indexes have been written out to the media if no message DSNU473 or DSNU479 were issued earlier. A SYSCOPY record of ICTYPE 'Q' has been recorded in the SYSCOPY catalog table for the table space and for each of its indexes with the COPY YES attribute. The START-RBA value in the SYSCOPY record(s) is the same for all objects quiesced during this invocation of the utility. This RBA/LRSN value will be printed in the message DSNU474 following this message. Processing continues to the next table space in the table space list, if any.

DSNU478I *csect-name* **- INDEX** *index-name* **HAS PENDING STATE**

Explanation: Index 'index-name' has a PENDING state.This message is repeated for all PENDING stated indexes in a table space.

Severity: 8 (error)

System action: QUIESCE stops processing but continues to check all table spaces and indexes for any additional table space or index that has a PENDING state. There is no update in SYSCOPY table for any table space in the list.

User response: Issue the DISPLAY DATABASE command to look at the utility status byte in the DBA table for a specified PENDING state.

Problem determination: Refer to Part 4 (Volume 1) of *DB2 Administration Guide* for a discussion of the database recovery function.

DSNU479I *csect-name* **- UNABLE TO EXTERNALIZE INDEX** *index-name* **PAGES**

Explanation: The attempt to externalize index 'index-name' pages to the media has failed.

Severity: 4 (warning)

System action: QUIESCE still continues but the table space cannot be volume-dumped.

Problem determination: The problem may be an I/O error.

DSNU480I *csect-name* **- UNABLE TO EXTERNALIZE TABLESPACE** *tablespace-name* **PARTITION** *partition-number* **PAGES**

Explanation: The attempt to externalize TABLESPACE 'tablespace-name' PARTITION 'partition-number' pages to the media failed.

Severity: 4 (warning)

System action: QUIESCE still continues, but the table space cannot be volume-dumped.

Problem determination: The problem might be an I/O error.

DSNU481I *csect-name* **- QUIESCE SUCCESSFUL FOR** *obj-type obj-qual.obj-name* **PARTITION** *partition*

Explanation: The QUIESCE for PARTITION 'partition-number' of this object was successful. If the object is a table space partition, then the quiesce for all of its indexes was successful. This message will also be issued for each index with the COPY YES attribute over the quiesced table space. This message is repeated for every quiesced table space in the list.

Severity: 0 (informational)

System action: All data held in the DB2 buffer pool(s) for the table space and its indexes have been written out to the media if message DSNU480 or DSNU482 was not issued earlier. A SYSCOPY record of ICTYPE 'Q' was recorded in the SYSCOPY catalog table for the table space and for each of its indexes with the COPY YES attribute. The START-RBA value in this SYSCOPY record is the same for all objects quiesced during this invocation of the utility. This RBA value is printed in message DSNU474 following this message. Processing continues to the next table space in the table space list, if any.

DSNU482I *csect-name* **- UNABLE TO EXTERNALIZE INDEX** *index-name* **PARTITION** *partition-number* **PAGES**

Explanation: The attempt to externalize index 'index-name' 'partition-number' pages to the media failed.

Severity: 4 (warning)

System action: QUIESCE still continues, but the table space cannot be volume-dumped.

Problem determination: The problem might be an I/O error.

DSNU483I *csect-name* **- TABLESPACE** *tablespace-name* **PARTITION** *partition-number* **HAS WRITE ERROR RANGE OR IS IN DEFERRED RESTART.**

Explanation: The attempt to externalize table space 'tablespace-name' partition 'partition-number' pages to the media failed.

Severity: 4 (warning)

System action: QUIESCE still continues, but you cannot make an image copy of the table space or partition. Processing continues for the index partition and all nonpartitioned indexes within the table space, and then proceeds to the next table space in the table space list, if any.

User response: Issue DISPLAY DATABASE to determine what exception state, if any, would prevent image copy on the table space or partition.

Problem determination: Refer to Part 4 (Volume 1) of *DB2 Administration Guide* for a discussion of the database recovery function.

DSNU484I *csect-name* **- INDEX** *index-name* **PARTITION** *partition-number* **HAS WRITE ERROR RANGE OR IS IN DEFERRED RESTART**

Explanation: The attempt to externalize index 'index-name' pages partition 'partition-number' pages to the media failed.

Severity: 4 (warning)

System action: QUIESCE processing continues.

User response: Issue DISPLAY DATABASE to determine the exception state.

Problem determination: Refer to Part 4 (Volume 1) of *DB2 Administration Guide* for a discussion of the database recovery function.

DSNU500I *csect-name* **RECOVERY COMPLETE, ELAPSED TIME =** *hh/mm/ss*

Explanation: This message is issued at the normal completion of the RECOVER utility. The time that elapsed since the beginning of the RECOVER utility to its completion was 'hh' hours, 'mm' minutes, and 'ss' seconds.

Severity: 0/4 (informational/warning)

System action: This message indicates the normal completion of the RECOVER utility. Processing continues to the next utility statement in the utility job step, if any. If not, the utility job step is normally terminated, and the resources held become available for further processing. The UTILID of the utility job step becomes available for reuse.

User response: The recovered data is now available for further processing.

Operator response: The recovered resources are available for further processing.

DSNU501I *csect-name* **A PAGE HAS BEEN STOPPED DURING RECOVERY. SEE PREVIOUS DSNI012 MESSAGE**

Explanation: This message is issued at the normal completion of the RECOVER utility if the data manager subcomponent cannot apply all the log records for the page because of an inconsistent DB2 log.

Severity: 8 (error)

System action: This message indicates that the RECOVER utility completed normally except for the page (or pages) that was indicated by the previous DSNI012 messages. The data manager subcomponent marked the pages as broken. All other pages were recovered successfully. Log apply was stopped at the point of the inconsistency in the DB2 log for that page.

User response: For information about using the REPAIR utility to apply changes that should have been made by the log record in error, refer to *DB2 Utility Guide and Reference*. If you were recovering to the current point in time, resubmit the RECOVER job, and specify 'PAGE(page-number) CONTINUE'. This causes DB2 to retrieve the page from DASD, and apply log records starting after the faulty one.

DSNU502I *csect-name* **PAGE** *page number* **WAS NOT FOUND IN ANY IMAGE COPY DATA SET**

Explanation: The page specified for 'RECOVER page' was not found in any image copy data sets.

This is an informational message only. It does not indicate termination of processing.

Severity: 0 (informational)

System action: The RECOVER utility continues processing. The assumption is that the page was inserted after the last image copy was made. The page is reset and recovered from the log, starting at the log RBA recorded in the SYSCOPY record for the last image copy data set.

DSNU503I *csect-name* **PAGE** *page-number* **IS WITHIN PHYSICAL ERROR RANGE**

Explanation: The page specified for 'RECOVER page' is within the physical error range of the table space partition, index space partition, or data set. 'RECOVER page' cannot be effective in this case, because the page would still be within the physical error range and therefore inaccessible to the user.

Severity: 8 (error)

System action: The RECOVER utility terminates with return code 8. No changes are made to the table space.

User response: Resubmit the RECOVER utility specifying 'error range'.

DSNU504I *csect-name* **MERGE STATISTICS FOR** *obj-type obj-qual.obj-name* **- NUMBER OF COPIES =** *cc*, **NUMBER OF PAGES MERGED =** *nnnnnn*, **ELAPSED TIME =** *hh/mm/ss*

Explanation: This message is issued at the completion of the MERGE process for the MERGECOPY or RECOVER utility. The message indicates the number of image copies currently processed ('cc') and the total number of pages that were merged ('nnnnnn'). The same page can be obtained from more than one incremental image copy. MERGE is repeated for MERGECOPY if all of the input image copy data sets cannot be allocated and a temporary work data set has been specified.

For the RECOVER utility, MERGE is executed whether or not there are incremental image copies participating. In this case, the number of copies ('cc') is one.

The elapsed time of the MERGE is 'hh/mm/ss', hours, minutes, and seconds, respectively.

System action: The MERGECOPY or RECOVER utility continues processing; this message is for information only.

User response: The statistics are indicative of the processing required. For MERGECOPY, it could indicate that next time more input image copy data sets should be allocated. For example, if there was more than one MERGE pass then this message will be issued more than once. For both MERGECOPY and RECOVER, it could indicate that more frequent invocation of MERGECOPY is appropriate.

DSNU505I *csect-name* **PAGE** *page-number* **IS BEYOND END OF** *obj-type*

Explanation: A page number was found to be beyond the end of the table space, index space, or the current data set.

Severity: 8 (error)

System action: The RECOVER utility terminates with ABEND 04E reason code 00E40100. If RECOVER PAGE was specified, then the utility terminates with return code 8 and no changes are made to the object.

User response: Check that the image copy data set(s) in use during the RECOVER contain valid pages. If running RECOVER with the PAGE option, then resubmit the utility with the correct page specified.

DSNU506I *csect-name* **TABLESPACE** *obj-qualifier.obj-name* **DSNUM** *dsnum* **IS IN REORG PENDING STATE**

Explanation: This message is issued at completion of a utility when an object has the REORP state set.

The utility can set the REORP state if the table space contains a table that includes an identity column. It can also set the REORP state when a partitioned table space, which has had its limiting keys altered, is recovered to a point in time.

Severity: 0 (informational)

System action: The utility continues processing.

DSNU507I *csect-name* **NO USABLE BACKUP FOR RECOVER**

Explanation: The RECOVER utility determined that a backup was required but not found. The utility encountered one of the following situations:

- An error occurred while processing the oldest available full image copy data set. This might be the only full image copy data set available.
- While scanning backward through the SYSCOPY table, a LOAD LOG(NO) or REORG log(NO) record type was found.

Any of the following is acceptable as a backup:

- A full image copy.
- A LOAD LOG(YES).
- A REORG LOG(YES).
- For RECOVER PAGE, an incremental image copy that contains the specified page.
- A 'no record found' return from the first request for a SYSCOPY record indicates that the object was entirely populated by 'inserts'. The object will be recovered from the DB2 log if its entire history lies within the scope of the log, that is, archive log data sets containing updates for the object have not been deleted.

Severity: 8 (error)

System action: The RECOVER utility terminates.

User response: Re-create the object that was to be recovered.

DSNU508I *csect-name* **IN FALL BACK PROCESSING TO PRIOR FULL IMAGE COPY**

Explanation: One of the following occurred:

- The RECOVER utility could not allocate the most recent full image copy data set. The prior full image copy is used.

  Accompanying messages indicate why the image copy data set could not be allocated.
- PAGE or ERROR RANGE was specified and the recoverable point identified an image copy that was created by the DFSMSdss DUMP command with the CONCURRENT option. Such an image copy cannot be used to process PAGE or ERROR RANGE. A prior recoverable point is located.

System action: The RECOVER utility continues processing, searching for the prior image copy.

DSNU509I *csect-name* **- NO ERROR RANGE TO RECOVER**

Explanation: The RECOVER utility was invoked to recover the existing ERROR RANGE. However, no ERROR RANGE was found.

Severity: 4 (warning)

System action: This message indicates the completion of the RECOVER ERROR RANGE utility. Processing continues to the next utility statement in the utility job step. If there are no more utility statements, the utility job step is normally terminated.

DSNU510I *csect-name* **NO GOOD FULL IMAGE COPY DATA SET FOR RECOVERY OF** *obj-type obj-qual.obj-name*

Explanation: It is not possible to locate a good image copy data set from which to recover. Either no back level image copy exists, an I/O error occured while reading the back level image copies, or DB2 was unable to locate a suitable image copy. The return code that accompanies this message is 0 or 8 under the following conditions:

- The return code is 0 if a full image copy is not found in the SYSCOPY table during RECOVER initialization or during fall back processing. However, a LOG(YES) event was detected and recover continues from the log. Note that if you have performed a REORG after your last image copy, then no full image copy may be found in the SYSCOPY table.
- The return code is 8 if the RECOVER utility is in fall back mode and no base recoverable point (a full image copy or LOG(YES) event) was found in the SYSCOPY table. Further utility processing is terminated.

Severity: 0 (informational) or 8 (error)

System action: If the return code is 0, processing continues. If the return code is 8, further utility processing is terminated.

User response: Notify the system programmer if the return code is 8. If the return code is 0, no action is required.

Operator response: If the return code is 8, the object in question should be started for utility-only processing (if it is not in this state already) until corrective action is taken. If the return code is 0, no action is required.

System programmer response: If the return code is 8 and the object being recovered is a table space, the data must be repaired so that an image copy can be taken. An internal error might be involved, and system support might be required. If the return code is 8 and the object being recovered is an index space, then the REBUILD INDEX utility can be used to rebuild the index from the data. If the return code is 0, no action is required. Refer to Part 4 (Volume 1) of *DB2 Administration Guide* for a discussion of database recovery procedures.

DSNU511I *csect-name* **RECOVERY INFORMATION INDICATES THAT** *obj-type obj-qual.obj-name* **IS NOT IN A RECOVERABLE STATE**

Explanation: The requested recovery of an object would result in recovery entirely from the log. Because the object is also in copy pending status, recovery might fail in log apply in these situations:

- A previous MODIFY utility job deleted the last recoverable point indicated in the SYSCOPY catalog table.
- A previous MODIFY utility job deleted all SYSCOPY entries.

In either case, the MODIFY utility leaves the object in copy pending status and the RECOVER utility is unable to determine the reliability of recovery from the log.

Severity: 8 (error)

System action: The RECOVER utility terminates with return code 8.

User response: Determine the recoverability status of the object. SYSCOPY information is probably not useful as no recoverable point (FULL IMAGE COPY, REORG LOG(YES) or LOAD REPLACE LOG(YES)) was found by the RECOVER utility. User information might be the only indication about recoverability.

To allow RECOVER to execute, copy pending status must be turned off. It is safest to make a full image copy before attempting the recovery. Then run the RECOVER utility with the TORBA or TOLOGPOINT option.

If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 5.

DSNU512I *csect-name* **DATA SET LEVEL RECOVERY IS REQUIRED FOR** *obj-type obj-qual.obj-name*

Explanation: Recovery was requested for an entire object. However, there is an image copy made for a specific data set associated with the object. Hence, the object must be recovered by data set.

Severity: 8 (error)

System action: RECOVER completes unsuccessfully with a return code of 8.

User response: Terminate the current RECOVER job. Then, submit a RECOVER job by specifying 'DSNUM(data set number)' for each data set for which recovery is required. Refer to *DB2 Utility Guide and Reference* for a discussion of RECOVERY utility constraints.

DSNU513I *csect-name* **RECOVER UTILITY LOG APPLY RANGE IS RBA** *xxx* **LRSN** *xxx* **TO RBA** *yyy* **LRSN** *yyy*

Explanation: The RECOVER utility issues this message one or more times to display the entire log range before applying updates. The RBA and LRSN values are equal in a non-data sharing environment. The RBA values will be zero in a data sharing environment for the catalog and directory objects that do not have DSNDB01.SYSLGRNX entries. This includes catalog table spaces DSNDB06.SYSCOPY and DSNDB06.SYSGROUP and all of their indexes, and all of the directory objects.

Severity: 0 (informational)

System action: The RECOVER utility continues processing.

DSNU514I *csect-name* **RECOVERY DATA DOES NOT PERMIT DSNUM(ALL) RECOVERY OF** *obj-type obj-qual.obj-name*

Explanation: Recovery was requested for an entire object. If the object is a table space, then the latest activity on the entire table space was a REORG LOG (NO) or a LOAD LOG (NO). If the object is an index space, then the latest activity on the entire index space was a REORG INDEX or REBUILD INDEX. If full image copies have been taken at the data set level since then, recover the object on the data set level.

Severity: 8 (error)

System action: RECOVER completes unsuccessfully with a return code of 8.

User response: Submit a recover job by specifying DSNUM (data set number) for each data set requiring recovery. If the object is an index space, then the REBUILD INDEX utility may be used to rebuild the index. Refer to *DB2 Utility Guide and Reference* for a discussion of RECOVERY utility constraints.

DSNU515I *csect-name* **THE IMAGE COPY DATA SET** *dsn* **WITH DATE=***date* **AND TIME=***time* **IS PARTICIPATING IN RECOVERY OF** *obj-type obj-qual.obj-name*

Explanation: This message is issued by the RECOVER utility for each participating image copy data set before it opens the data set and begins the restore process. If an error occurs while processing, the utility searches for a new recoverable base.

This message is routed to SYSOUT.

Severity: 0 (informational)

System action: RECOVER processing continues.

User response: Refer to *DB2 Utility Guide and Reference* for a discussion of RECOVERY utility constraints.

DSNU516I *csect-name* **NO FULL IMAGE COPY AFTER LOAD, REORG OR CHECK DATA LOG(NO) OF** *obj-type obj-qual.obj-name*

Explanation: RECOVERY was requested for a data set of an object. If the object is a table space, then a full image copy was not taken after or during a LOAD, REORG, or CHECK DATA without logging. The table space is in an unrecoverable state. If the object is an index space, then a full image copy was not taken after a REORG INDEX or REBUILD INDEX.

Severity: 8 (error)

System action: RECOVER completes unsuccessfully with a return code of 8.

User response: If the object is a table space, the data set is in an unrecoverable state. If the object is an index space, then the REBUILD INDEX utility may be used to rebuild the index. Notify the system programmer for corrective action.

System programmer response: The data should be repaired before a full image copy is taken. The recovery information for the object (use the REPORT RECOVERY utility) should be scrutinized. Refer to Part 4 (Volume 1) of *DB2 Administration Guide* for a discussion of database recovery procedures.

DSNU517I *csect-name* **- SYSCOPY RECORD DELETED BY MODIFY UTILITY. DSN=***dsn***, VOL=SER=***nnn***, FILESEQNO=***fff***

Explanation: This message is repeated for each volume of a multivolume data set that is deleted.

The variables have the following meanings:

dsn The name of the image copy data set whose record was removed from the SYSCOPY table by the MODIFY utility.

nnn The serial number of the volume containing the data set.

fff The tape file sequence number. It will be zero for DASD data sets.

Severity: 0 (informational)

System action: The MODIFY utility continues its termination processing.

System programmer response: Because DB2 no longer has any knowledge of the image copy data set, you may delete it and reuse the space. If the last full image copy was deleted, ensure that another one is taken as soon as possible.

DSNU518I *csect-name object-type object-qualifier.object-name* **CONTAINS BROKEN PAGE** *page-number***, ERROR CODE** *error*

Explanation: A broken page, *page number*, was detected during execution of an image copy utility job. The copy step is terminated and the broken page is not copied to the output data set.

Severity: 8 (error)

System action: COPY utility ends with a return code of 8.

User response: The associated data set has not been copied. If the object is in the COPY-pending status, it is in an unrecoverable state. Notify the system programmer for corrective action.

System programmer response: Use either the RECOVER, RECOVER PAGE, or REPAIR utility to correct the broken page. Refer to Part 4 (Volume 1) of *DB2 Administration Guide*. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNU519I *csect-name* **TOCOPY DATA SET NOT FOUND**

Explanation: The data set specified with the keywords TOCOPY, TOVOLUME, or TOSEQNO cannot be found in the SYSCOPY table for the object to be recovered.

This may be caused by a mismatch in granularities. If the RECOVER specifies a particular DSNUM, the data set specified with the TOCOPY keyword may be for an image copy of this DSNUM or of the entire table space or index space. However, if the RECOVER is for the entire table space or index space, the specified data set must be for an image copy of the entire table space or index space.

This message may also be issued if TOVOLUME is specified for a cataloged data set.

RECOVER TOCOPY can issue DSNU519I even if the data set information was correct but the data set was not for this site. For example, if the DB2 subsystem is specified as LOCALSITE, but the TOCOPY data set

information is for a RECOVERYSITE data set, this message will be issued.

Severity: 8 (error)

System action: RECOVER processing is terminated.

User response: Correct the input and resubmit the RECOVER job.

DSNU520I *csect-name* **RECOVER TO A PRIOR POINT IN TIME MIGHT LEAVE TABLE SPACE** *table-space-name* **INCONSISTENT**

Explanation: RECOVER DSNUM to a previous point in time (for example, using the TOLOGPOINT option) was specified for a nonpartitioned table space. You might have recovered a data set that has different compression characteristics than the replaced data set. For example, the data set to which you recovered might have been compressed with a different dictionary.

Severity: 4 (warning)

System action: Processing continues.

User response: The table space might be in an inconsistent state. If the dictionary was not built with the data set that was just recovered, the results are unpredictable and might jeopardize all the data in the entire table space.

Recover all of the data sets in the table space to the same point in time. Do not ignore this message unless all data sets in this table space were recovered to the same point in time. Otherwise, the state of this nonpartitioned table space is inconsistent and unpredictable.

DSNU521I *csect-name* **- RECOVER UTILITY FAILED. UNABLE TO READ REQUIRED LOG RECORDS.**

Explanation: Recovery was requested for a table space, index space, or data set. An attempt was made to apply changes from the DB2 log, but the necessary log records could not be read from the log data set. This message is accompanied by message DSNI007I.

Severity: 8 (error)

System action: The RECOVER utility completes unsuccessfully with return code 8.

User response: Consult the system programmer for corrective action.

Operator response: Provide the console listing of the DSNI007I message for the system programmer.

System programmer response: Run the print log map (DSNJU004) utility. Find the data set that contains the requested RBA or LRSN from the output of the utility.

Problem determination: If no listed data set contains the required RBA or LRSN, one of the following might have occurred:

- The data set was deleted from the bootstrap data set (BSDS) by the change log inventory (DSNJU003) utility. Rerun the utility to insert the missing data set name. DB2 must be stopped to run this utility.
- The BSDS was damaged. The BSDS can be restored using the dump and the most recent archive log data set. Run the change log inventory (DSNJU003) tility to supply changes that occurred during and after the last offload operation. Changes can be determined from the operator's console listing.

If a listed log data set contains the required RBA or LRSN, this message is preceded by one or more DSNJxxx error messages. See the explanation of those messages for the appropriate action.

DSNU522I *csect-name* **RECOVER CANNOT PROCEED FOR** *obj-type obj-qual.obj-name* **BECAUSE A NON-RECOVERABLE EVENT HAS BEEN ENCOUNTERED FROM SYSIBM.SYSCOPY WHICH HAS DBNAME=***dbname* **TSNAME=***tsname* **DSNUM=***dsnum* **ICTYPE=***ictype* **START_RBA=X'***rba***'**

Explanation: Recovery was requested for an index, but activity on its table space (identified by the SYSIBM.SYSCOPY columns) prohibits recovery. The use of an index's full image copy by recovery is prohibited by a REORG or LOAD on its underlying table space.

Severity: 8 (error)

System action: Utility processing is terminated. All objects in the RECOVER statement are left in the same state as before the RECOVER utility was invoked.

User response: Use the REBUILD INDEX utility to rebuild the index from the table space. For more information about the recoverability of this index, run the REPORT RECOVERY utility on both the index and its table space.

DSNU523I *csect-name* **- RECOVER UTILITY FAILED. NO FULL IMAGE COPY WAS AVAILABLE AND THE FIRST LOG RECORD WAS NOT THE REQUIRED INITIALIZATION RECORD.**

Explanation: RECOVER was requested for a table space or a data set of a table space and no image copy was available. Recovery was attempted from the log and failed due to missing data set initialization records. Specifically, the log records which format space map and header pages were missing.

Severity: 8 (error)

System action: RECOVER completes unsuccessfully with a return code of 8.

User response: The associated data set has been reset to an empty state. Notify system programmer for corrective action.

System programmer response: Use the problem determination information given below to determine if a user error has occurred. If possible, correct the error and resubmit the RECOVER utility job for this table space or data set. Refer to Part 4 (Volume 1) of *DB2 Administration Guide* for a discussion of possible recovery procedures.

Problem determination: This message was preceded by message DSNU513I which identifies the log RBA range being applied. Run the print log map (DSNJU004) utility for the table space in question over the indicated log RBA range.

This will show what log records are present and may assist in determining if a log data set has been damaged or deleted, or if the BSDS has been damaged. If damage to table space DSNDB01.SYSLGRNX is suspected, recover it and attempt the earlier failing recovery again.

DSNU524I *csect-name* **INVALID DATA SET OR PARTITION NUMBER num SPECIFIED FOR** *obj-type obj-qual.obj-name*

Explanation: An invalid number was specified for the data set or partition of the object named in the utility statement.

Severity: 8 (error)

System action: Further utility processing terminates.

User response: Correct the data set/partition number, or verify that the object name is specified correctly. Then resubmit the utility statement in a new utility job step. Refer to *DB2 Utility Guide and Reference* for a description of the particular utility options.

DSNU525I *csect-name* **RECOVERY DATA DOES NOT PERMIT DSNUM(ALL) RECOVERY OF** *obj-type obj-qual.obj-name*

Explanation: Recovery was requested for an entire object. However, the object is not recoverable at that level for one of the following reasons:

- The object was recovered to a point in time (for example, TOLOGPOINT was specified), which required fall back processing. However, no subsequent image copy was taken to establish a base for recovery.
- A partition of the object was recovered to a point in time after the last full image copy of the object was taken.

Severity: 8 (error)

System action: The RECOVER utility completes unsuccessfully with return code 8.

User response: Submit the RECOVER job with DSNUM specified for each data set for which recovery is required. Refer to *DB2 Utility Guide and Reference* for information about the RECOVER utility.

DSNU526I *csect-name* **- NO FULL IMAGE COPY AFTER PARTIAL RECOVERY OF** *obj-type obj-qual.obj-name*

Explanation: Recovery was requested for a data set of an object. However, an earlier point-in-time recovery (for example, using the TOLOGPOINT option) was specified and no subsequent image copy was taken to establish a base for recovery. The object is in a nonrecoverable state.

Severity: 8 (error)

System action: The RECOVER utility completes unsuccessfully with return code 8.

User response: The associated data set is in a nonrecoverable state. Consult the system programmer for corrective action.

System programmer response: You can use the RECOVER utility with the TOCOPY, TORBA, or TOLOGPOINT option to force recovery to an earlier point in time. You can use the DSN1COPY utility to restore the data set to an earlier image copy. Refer to Part 4 (Volume 1) of *DB2 Administration Guide* for a discussion of database recovery procedures.

DSNU527I *csect-name* **DB2 OBJECT RECOVERED**
TO AN INDETERMINATE POINT

Explanation: The image copy data set specified with
the TOCOPY keyword of the RECOVER utility was one
of the following:

• An image copy dataset made with the SHRLEVEL
CHANGE option. In this mode, changes might have
been made to the object while image copy was in
progress. If a change was made to a page before
image copy processed the page, the update will be in
the copy data set. If a change was made to a page
after image copy processed it, the change will not
appear in the copy data set. The data set may
contain partial updates from some transactions.

• An image copy dataset made during the LOAD
REPLACE utility. An image copy produced during
LOAD REPLACE will not contain updates made to
the table space to correct unique index violations
during the INDEXVAL phase or referential constraint
violations during the ENFORCE phase. Therefore, it
is not recommended as a recovery point.

If the object recovered is a table space, then indexes on
the table space are placed in rebuild pending status and
dependent table spaces, if any, are placed in check
pending status. The indexes placed in rebuild pending
are identified by message DSNU831I, and the

dependent table spaces placed in check pending are
identified by message DSNU563I.

Severity: 4 (warning)

System action: The utility terminates normally.

User response: Based on the explanation above,
determine if the recovery is satisfactory. Rebuild the
indexes identified by message DSNU557I in the job
output. Run CHECK DATA on the table spaces
identified by message DSNU563I to correct any
referential constraint violations.

Refer to Part 4 (Volume 1) of *DB2 Administration Guide*
for a discussion of database recovery procedures.

DSNU528I **NO FULL IMAGE COPY WAS AVAILABLE AND THERE ARE NO UPDATES TO APPLY FROM THE DB2 LOG FOR TABLESPACE** *tablespace-name* **DSNUM** *dsnum*

Explanation: Recovery was requested for a table space or a data set of a table space and no image copy was available. If *dsnum* is 0, then recovery was requested at the table space level. Recovery was attempted from the log, but no log records were applied because no SYSLGRNX records corresponding to the recovery scope were found.

Severity: 8 (error)

System action: The associated data set was reset to an empty state and the data set was placed in REBUILD PENDING status. If REBUILD was processing a TABLESPACE list, only the associated data set was reset to an empty state but all data sets in the list were placed in REBUILD PENDING status.

User response: If the TORBA or TOLOGPOINT option was specified, use the information in the problem determination section to verify that it is a valid RBA or LRSN. Correct the error and resubmit the REBUILD utility for the table space.

System programmer response: Use the information in the problem determination section to determine if a user error occurred. Correct the error and resubmit the REBUILD utility for the table space. Also refer to Part 4 (Volume 1) of *DB2 Administration Guide* for a discussion of possible recovery procedures.

Problem determination: Execute the REPORT RECOVERY utility for the table space at the appropriate table space or data set level (that is, DSNUM). The output shows the recovery points (SYSCOPY records) and the update log ranges (SYSLGRNX records) that exist for the table space.

DSNU529I *csect-name* **INVALID SPECIFICATION OF TORBA =** *xxx*

Explanation: 'xxx' is the log RBA specified for the RECOVER utility TORBA keyword. The RBA is not within the scope of the log.

Severity: 8 (error)

System action: The RECOVER utility terminates with a return code of 8 without making any changes to the object to be recovered.

User response: Correct the input and resubmit the job. Refer to Part 4 (Volume 1) of *DB2 Administration Guide* for a discussion of database recovery procedures.

Problem determination: The listing produced by the print log map (DSNJU004) utility will show the current extent of the DB2 log.

DSNU530I *csect-name* **- MULTIPLE DATA SETS FOUND FOR THE COPY DATA SET SPECIFICATION**

Explanation: DB2 issues this message during the RECOVER or COPYTOCOPY utility when the specification of the input image copy dataset is ambigous and does not uniquely identify an image copy. Multiple image copies were found in SYSIBM.SYSCOPY for the object with the same data set name.

Severity: 8 (error)

System action: Processing halts for this object.

RECOVER completes unsuccessfully with a return code of 8.

For the COPYTOCOPY utility, if the utility job step includes an OPTIONS control statement with the EVENT(ITEMERROR,SKIP) option, the utility will process the next item in the list.

User response: Two or more DSNU531I messages will follow this message providing more information about each duplicate data set. Select the correct data set from the DSNU531I messages.

Rerun the RECOVER job, specifying a unique combination of TOVOLUME and TOSEQNO.

Rerun the COPYTOCOPY job, specifying an unique combination of FROMVOLUME and FROMSEQ.

DSNU531I *csect-name - dsn* **EXISTS WITH - VOLUME=***vol***, FILE SEQUENCE NUMBER=***fseq***, RBA=***rba***, DATE=***date***, TIME=***time***

Explanation: DB2 issues this message two or more times to document the duplicate data sets detected by the RECOVER utility during TOCOPY processing, or by the COPYTOCOPY utility during FROMCOPY procesing. See the explanation for message DSNU530I.

Severity: 8 (error)

System action: RECOVER completes unsuccessfully with a return code of 8.

For COPYTOCOPY, if the utility job step includes an OPTIONS control statement with the EVENT(ITEMERROR,SKIP) option, the next item in the list will be processed.

User response: Use the information provided in all of the DSNU531I messages to determine the correct data set for this recovery. 'rba', 'date' and 'time' are provided for each data set to assist in the selection. The DSNU531I message for the data set you select will provide the correct 'vol' and 'fseq' values to be specified on the TOVOLUME and TOSEQNO keywords respectively.

Rerun the RECOVER job specifying TOCOPY and TOVOLUME(vol-ser). Include the TOSEQNO(fseq) keyword if it is necessary to distinguish an unique data set.

Rerun the COPYTOCOPY job specifying FROMCOPY and FROMVOLUME(vol-ser). Include the FROMSEQNO(fseq) keyword if it is necessary to distinguish a unique data set.

DSNU532I *csect-name* **- RECOVER** *obj-type obj-qual.obj-name* **START**

Explanation: This message is issued when there is more than one object in the list. It indicates that the recover of a particular object is starting. If the PARALLEL keyword has been specified, then the restore phase messages for different objects in the list may be intermingled.

This message is routed to SYSPRINT.

Severity: 0 (informational)

System action: RECOVER processing continues.

DSNU533I *csect-name* **-** *obj-type obj-qual.obj-name* **IS A DUPLICATE OF** *obj-type obj-qual.obj-name*

Explanation: The utility found a duplicate object in the list. The object name may indicate a partition or piece number.

Severity: 8 (error)

System action: The utility job terminates processing.

User response: Correct the object list. Resubmit the job if needed.

DSNU534I *csect-name* **TARGET RECOVERY POINT IS NOT A QUIESCE POINT**

Explanation: The RECOVER utility attempted to recover to a point in time (for example, using the TOLOGPOINT option) and one of the following conditions has been detected:

- The table space set is not being recovered to a quiesce point or to a common image copy SHRLEVEL(REFERENCE) point.
- Not all table spaces in the table space set have the same quiesce point.
- A table space in the set contains a dependent table with a referential constraint that was defined after the specified quiesce point.

A table space set is a group of table spaces that are referentially related or it can be a base table space and its associated LOB table spaces.

Severity: 4 (warning)

User response: If possible, recover the table space set to a quiesce point or a common image copy SHRLEVEL(REFERENCE) point. Otherwise, use the CHECK DATA utility to check and correct referential violations. Use the CHECK DATA utility on the base table space and the CHECK LOB utility on the LOB table space(s) to validate the LOB columns.

DSNU535I *csect-name* **- FOLLOWING TABLESPACES RECOVERED TO A CONSISTENT POINT**

Explanation: This message will be issued by the RECOVER utility when it determines that a table space set (a group of referentially related table spaces or a base table space and its lob table spaces) were recovered to a quiesce or to a common image copy SHRLEVEL(REFERENCE) point. All table spaces in the table space set will be listed after this message heading.

Severity: 0 (informational)

System action: No action is taken.

DSNU536I *csect-name* **- RELATIONSHIP** *r* **ON TABLE** *t* **CREATED AFTER QUIESCE POINT**

Explanation: The relationship named 'r' was created on the dependent table, 't', after the quiesce point used for recovering to a point in time. The table space in which 't' resides is set to check pending beginning with the first record of 't'.

Severity: 4 (warning)

System action: No action is taken.

User response: Refer to Part 2 (Volume 1) of *DB2 Administration Guide* for the failure analysis procedure for the check pending state.

DSNU537I *csect-name obj-type obj-qual.obj-name* **IS NOT ALLOWED IN THE LIST**

Explanation: The RECOVER or COPY utility found a directory or catalog table space in a list with other table spaces or with user-defined indexes.

The table spaces that the COPY utility will not allow in a list with other table spaces are DSNDB01.SYSUTILX, DSNDB06.SYSCOPY, and DSNDB01.SYSLGRNX. However, these table spaces may be specified with their IBM-defined indexes in a list.

You cannot recover more than one directory or catalog table space in a list. Directory and catalog table spaces must be recovered individually in a particular order because the recovery of some objects depends on information derived from others. The correct order is documented in Part 4 (Volume 1) of *DB2 Administration Guide*. IBM-defined indexes may be specified with their associated directory or catalog table space in a list.

Severity: 8 (error)

System action: The utility job terminates.

User response: Remove the directory and catalog object names from the list. Then submit the COPY or RECOVER job again.

DSNU538I RECOVER ERROR RANGE OF DSN=dataset name ON VOLUME=volser FROM CCHH=X'cccchhhh' TO CCHH=X'cccchhhh' CONTAINS PHYSICAL ERROR

Explanation: This message indicates that the error range involved in the error range recover 'dataset name' is the data set which has the physical error. The data set resides on DASD volume 'volser' and the error range is from track address X'cccchhhh' to track address X'cccchhhh'.

If the error range of a data set covers more than one extent or more than one volume, then there will be one message per extent per volume.

Severity: 8 (error)

System action: Physical I/O error is detected during error range recover, processing terminated with return code 8.

System programmer response: A DSNU086 message is issued to the console by DB2 while running error range recovery; this indicates that the physical error is not recovered. User must run ICKDSF or equivalent service utility to recover the physical error then re-run the error range recover again.

Problem determination: Use the information provided by this message to run ICKDSF and then run error range recover again.

DSNU539I *csect-name* **- RECOVER PROCEEDS AT THE** *site-type* **SYSTEM**

Explanation: The RECOVER process continues at the RECOVERY SITE system, and/or the TRACKER SITE system. DB2 issues this message for informational purposes only.

Severity: 0 (informational)

System action: The utility continues.

DSNU540I *csect-name* **DYNAMIC ALLOCATION FOR DATA SET** *data-set-name* **WILL WAIT FOR UNIT TO BECOME AVAILABLE**

Explanation: A request to dynamically allocate a data set for a BATCH UTILITY job was refused because a unit was not available. The request was repeated after setting an indicator in the dynamic allocation parameter list to cause the job to wait until a unit becomes available.

Severity: 0 (informational)

System action: Processing continues after a unit becomes available and is allocated.

DSNU541I UNABLE TO ALLOCATE THE PRIMARY COPY DATASET *data-set-name*, **BACKUP COPY IS USED**

Explanation: If RECOVER or MERGECOPY has an allocation error on the primary copy data set, it will locate the backup copy data set for the same system (local or recovery). The utility continues.

Severity: 0 (informational)

DSNU542I *csect-name* **- UNABLE TO OPEN THE PRIMARY COPY DATASET** *dataset-name*, **BACKUP COPY IS USED**

Explanation: If RECOVER or MERGECOPY has an open error in the primary copy data set, it will deallocate that copy data set and try to allocate the backup copy data set for that same system (local or recovery). The utility continues.

Severity: 0 (informational)

DSNU543I *csect-name* **- I/O ERROR READING PAGE** *page_number* **IN THE PRIMARY COPY DATASET** *dataset-name*, **BACKUP COPY IS USED**

Explanation: If RECOVER or MERGECOPY has an I/O error when reading a page, with the specified page number, from the primary copied data set, it will locate the backup copied data set in that same system (local or recovery) and continue reading pages starting from the page number in error.

Severity: 0 (informational)

DSNU544I *csect-name* **- UNABLE TO OPEN THE COPY DATASET** *dataset-name* **FOR THE RECOVERY OF** *obj-type obj-qualifier.obj-name*, **RC=** *reason-code*

Explanation: RECOVER was unable to open the data set 'dataset-name'. because of 'reason-code' from the open abend. The 'reason-code' is taken from field SDWAABCC in the SDWA.

Severity: 0 (informational)

System action: The RECOVER utility continues processing, searching for the prior image copy or applying log records.

DSNU545I *csect-name* **- I/O ERROR READING PAGE** *page-number* **IN THE COPY DATASET** *dataset-name* **DURING RECOVERY OF** *obj-type obj-qual.obj-name*

Explanation: An I/O error was detected while reading page 'page-number' from copy dataset 'dataset-name'. The message is issued when 1) The data set is the primary copy data set and there is no backup data set. 2) Open error is from the backup copy data set.

Severity: 0 (informational)

System action: The RECOVER utility continues processing, searching for the prior image copy or applying log records.

DSNU546I **LOGONLY RECOVER CANNOT PROCEED BETWEEN BEGINNING RBA '***xxxxxxxxxxxx***' AND ENDING RBA '***yyyyyyyyyyyy***' ON DSNUM** *dsnum* **BECAUSE A NON-RECOVERABLE EVENT HAS BEEN ENCOUNTERED FROM SYSIBM.SYSCOPY WHICH HAS DBNAME=***database-name* **TSNAME=***space-name* **DSNUM=***dsnum* **ICTYPE=***x* **START_RBA=X'zzzzzzzzzzzz'.**

Explanation: A nonrecoverable event (for example, LOAD LOG(NO), REORG LOG(NO)) occurred within the log range between the beginning log RBA or LRSN and ending log RBA or LRSN for the RECOVER LOGONLY utility.

This message is issued by the following CSECT: DSNUCASA

Severity: 8 (error)

System action: The RECOVER utility completes unsuccessfully with return code 8.

System programmer response: Find a proper prior point in time as a target recovery point and rerun the RECOVER utility job with the TORBA, TOCOPY, or TOLOGPOINT option. For a discussion of database recovery procedures, refer to Part 4 (Volume 1) of *DB2 Administration Guide* .

DSNU547I **LOGONLY** *action* **CANNOT PROCEED BECAUSE THE STARTING RBA=X'xxxxxxxxxxxx' IN THE HEADER PAGE PGNO=X'xxxxxx' FOR** *obj-type obj-qual.obj-name* **IS INVALID.**

Explanation: One of the following results occurred:

- The data set was restored from an earlier release backup that does not have a starting RBA or LRSN in the header page.
- The TORBA or TOLOGPOINT option is specified to an RBA or LRSN that is lower than the starting RBA or LRSN in the header page.

Severity: 8 (error)

System action: The RECOVER or RESTORE utility job terminates with return code 8.

System programmer response: Use the RECOVER or RESTORE utility without the LOGONLY option, if possible, or change the RBA or LRSN of the TORBA or TOLOGPOINT option to a valid RBA or LRSN value.

DSNU548I **DBID=X'xxxx' and PSID=X'xxxx' IN THE HEADER PAGE OF** *obj-type obj-qual.obj-name* **DO NOT MATCH THE DB2 CATALOG.**

Explanation: The DBID and PSID in the header page do not match the DB2 catalog. It is likely that the wrong data sets were restored.

Severity: 8 (error)

System action: The RECOVER utility job terminates with return code 8.

System programmer response: Find the correct data sets and rerun the job.

DSNU549I **RECOVER** *obj-type obj-qual.obj-name dsnum* **USES ONLY DB2 LOGS STARTING FROM LOGPOINT=X'xxxxxxxxxxxx'.**

Explanation: The RECOVER utility issues one message for each DB2 object that uses only DB2 logs to recover. In a non data-sharing environment, the LOGPOINT value is specified by an RBA; in a data sharing environment it is specified by an LRSN.

Severity: 0 (informational)

System action: RECOVER utility processing continues.

DSNU550I *csect-name* **INDEX NAME=** *index-name* **IS EMPTY**

Explanation: While attempting to REBUILD the indicated index, it was determined that the table upon which the index is based is empty.

Severity: 4 (warning)

System action: Utility processing proceeds to the next utility request, if any.

DSNU551I *csect-name* **INDEX NAME=** *index-name* **PART=***part-num* **IS EMPTY**

Explanation: While attempting to REBUILD the indicated index partition, it was determined that the partition upon which the specified index partition is based is empty.

Severity: 4 (warning)

System action: Utility processing proceeds to the next utility request, if any.

DSNU552I *csect-name* **RECOVER CANNOT PROCEED BECAUSE THE** *starting/ending* **LOGPOINT** *xxx* **IS NOT WITHIN THE RECOVERY RANGE OF** *xxx* **AND** *xxx* **FOR** *obj-type obj-qual.obj-name*

Explanation: The starting or ending LOGPOINT found for this recovery job is outside the available recovery information. A LOGPOINT value is an RBA or LRSN. The RECOVER utility cannot process this request.

Severity: 8

System action: The RECOVER utility terminates.

User response: Use the recovery information that is available from the REPORT utility. Execute the RECOVER utility within the available recovery range.

DSNU553I *csect-name* **INVALID PARTITION NUMBER=***nnnn* **SPECIFIED FOR INDEX NAME=***iiii*

Explanation: An invalid partition number *nnnn* was specified for index *iiii* in the utility statement.

Severity: 8 (error)

System action: Further utility processing is terminated.

User response: Correct the partition number or verify that the index name is specified correctly, and resubmit the utility statement in a new utility job step. Refer to *DB2 Utility Guide and Reference* for a description of the particular utility options.

DSNU554I *csect-name* **- DUPLICATE INDEX NAMES EXIST IN THE SPECIFIED INDEX NAME LIST**

Explanation: When attempting to CHECK or REBUILD the indicated indexes, it was determined that duplicate index names were specified in the index name list.

Severity: 8 (error)

System action: The utility job terminated.

User response: Correct the index-name(s) in the index name list and resubmit the job.

DSNU555I *csect-name* **UNLOAD PHASE STATISTICS, NUMBER OF RECORDS=***xxxx*

Explanation: This message is issued at the normal completion of the UNLOAD phase of the REBUILD INDEX utility. It gives the number of records that were unloaded (xxxx) for the specified table space.

Severity: 0 (informational)

DSNU556I *csect-name* **RECOVER CANNOT PROCEED FOR** *obj-type obj-qual.obj-name* **BECAUSE A SYSCOPY RECORD HAS BEEN ENCOUNTERED WHICH HAS DBNAME=** *dbname* **TSNAME=** *tsname* **DSNUM=** *dsnum* **ICTYPE=** *ictype* **STARTRBA=X'***rba***' LOWDSNUM=** *lowdsnum* **HIGHDSNUM=***highdsnum***.**

Explanation: DB2 has encountered a SYSCOPY record that prohibits recovery processing. The reason for this error is listed in the ICTYPE column. The following list supplies the ICTYPE and its explanation:

R,S,W, or X
A SYSCOPY record for a REORG or LOAD REPLACE that redistributed data across partitions has been encountered. The partitions that were affected by the ALTER INDEX that changed the limit keys must be recovered in the same list. An STYPE=S indicates that the CATENFM utility was run, and a point-in-time recovery prior to the CATENFM process is not allowed unless the entire system (including the DB2 catalog and directory) are recovered to the same point.

A
An attempt has been made to recover an index to a point-in-time prior to a first ALTER TABLE COLUMN that altered the index key column. This attempt is not allowed.

- An STYPE=C indicates that an ALTER took place that added a column to a table and the column was not added to the index in the same commit scope. The index should be rebuilt rather than recovered.
- An STYPE=N indicates that the index was ALTERed to have the NOT PADDED attribute. The index should be rebuilt rather than recovered.
- An STYPE=P indicates that the index was ALTERed to have the PADDED attribute. The index should be rebuilt rather than recovered.
- An STYPE=R indicates that a table space partition was rotated and a point-in-time

recovery prior to the rotation is not allowed. If the object being recovered is an index partition or a LOB table space belonging to the rotated partition, then those objects cannot be recovered prior to the rotation.

- An STYPE=V indicates that an ALTER TABLE COLUMN occurred on an indexed column for a numeric data type change. The index should be rebuilt rather than recovered.
- An STYPE=Z indicates that a column that is in the key of an index that was versioned prior to DB2 Version 8 was altered.

F An attempt has been made to recover an index to a prior point-in-time and more than sixteen versions of ALTER TABLE COLUMN have been done to alter the index key. The point-in-time recovery is not allowed. The index should be rebuilt.

P An index is being recovered to a point-in-time that is different than that of its underlying table space. The recover of the index is not allowed. Use the REBUILD INDEX utility instead.

Severity: 8 (error)

System action: The Recover utility terminates.

User response: Run the REPORT RECOVERY utility, and analyze the report of the recovery information for the object.

If a SYSCOPY record for a REORG or LOAD REPLACE that redistributed partition data was encountered, then add the missing partitions to the recover list using the LOWDSNUM and HIGHDSNUM values as a guide.

If the object is an index, then use the REBUILD INDEX utility to rebuild the index.

DSNU557I *csect-name* **-** *obj-type obj-qual.obj-name* **IS IN RECOVER PENDING**

Explanation: If the REBUILD utility issued this message, the corresponding table space to this index was rebuilt to a point in time with keyword TOCOPY, TOLOGPOINT, or TORBA. Thus, the index was placed in REBUILD pending status.

If the REBUILD INDEX utility issued this message, the index is in REBUILD pending status. Either duplicate index entries existed during the BUILD phase of the REBUILD INDEX, or invalid KEY/RIDs, which do not belong to the partition of the cluster index, were detected. The duplicate index entries or invalid KEY/RID entries must be resolved before starting the REBUILD INDEX utility again to REBUILD the index and remove the index from REBUILD pending status.

If the REORG utility with SHRLEVEL REFERENCE or CHANGE (or -TERM UTILITY) issued this message, the index is in REBUILD pending status. During the renaming of a data set during REORG, either a data set with the source name did not yet exist or a data set with the target name already existed.

Severity: 4 (warning) or 8 (error)

User response: If the REBUILD or REORG utility (or -TERM UTILITY) issued this message, use the REBUILD INDEX utility to REBUILD this index.

If the REBUILD INDEX utility issued this message, take one of the following actions:

- Resolve the duplicate index entries or invalid keys and rerun the REBUILD INDEX utility.
- Use the REPAIR SET NORCVRPEND utility or the START DATABASE FORCE command to reset the REBUILD pending status. However, these two options will not correct the index.

DSNU558I *csect-name* **ALL INDEXES OF TABLESPACE=** *table space-name* **ARE IN REBUILD PENDING**

Explanation: The REBUILD TORBA, TOLOGPOINT, or TOCOPY utility issues this message to inform the user of the state of all indexes belonging to the REBUILDed table space. All indexes were put in REBUILD pending status to prevent data and index inconsistency. All indexes should be REBUILDed before they are used.

System action: The REBUILD utility completed successfully.

Severity: 4 (warning)

User response: REBUILD or rebuild all indexes of the REBUILD table space.

DSNU559I *csect-name* **-** *obj-type obj-qual.obj-name* **PARTITION** *partition-number* **IS IN RECOVER PENDING STATE**

Explanation: If the REBUILD utility issued this message, one partition of a partitioned table space is being rebuilt to a point in time with keyword TOCOPY, TOLOGPOINT, or TORBA. The corresponding index for that partition was placed in rebuild pending status.

If the REBUILD INDEX utility issued this message, the index or the partition of a cluster index is in rebuild pending status. Either duplicate index entries existed during the BUILD phase of the REBUILD INDEX or INDEX PART, or invalid KEY/RIDs, which do not belong to the partition of the cluster index, were detected. The duplicate index entries or invalid KEY/RID entries must be resolved before starting the REBUILD INDEX utility again to rebuild the index and remove the index from rebuild pending status.

If the REORG utility with SHRLEVEL REFERENCE or CHANGE (or -TERM UTILITY) issued this message, one partition of a partitioned index was placed in rebuild pending status. During the renaming of a data

set during REORG, either a data set with the source name did not yet exist or a data set with the target name already existed.

Severity: 4 (warning) or 8 (error)

User response: If the REBUILD or REORG utility (or -TERM UTILITY) issued the message, rebuild this index with the REBUILD INDEX utility.

If the REBUILD INDEX utility issued this message, take one of the following actions:

- Resolve duplicate index entries or invalid keys and rerun the REBUILD INDEX utility.
- Use the REPAIR SET NORCVRPEND utility or START DATABASE FORCE command to reset the rebuild pending status. However, these two options will not correct the index.

DSNU560I *csect-name* **- TABLESPACE=** *database.tablespace* **PART=** *partition number* **IS IN RECOVERY PENDING STATE**

Explanation: The table space partition of a table space on which the utility was running is in recovery pending status. A RECOVER TABLESPACE at the partition level is required.

Severity: 4 (warning)

System action: No action is taken.

User response: Take one of the following actions:

- Run the RECOVER TABLESPACE utility on the table space partition.
- Run the LOAD REPLACE utility on the table space partition.
- Use the REPAIR SET NORCVRPEND utility or the START DATABASE FORCE command to reset the recovery pending status. However, these two options will not correct the table space.

DSNU561I *csect-name* **-** *obj-type obj-qual.obj-name* **PARTITION** *part-number* **IS IN CHECK PENDING**

Explanation: This message is issued to report the check pending state of the specified partition.

Severity: 4 (warning)

User response: If the object is a table space partition with referential relationships, then run the CHECK DATA utility to check for referential integrity violations.

If the object is a base table space partition, then run the CHECK DATA utility to validate the auxiliary columns.

If the object is an index space partition, then run the CHECK INDEX utility to verify that the index keys are consistent with the data in the table space partition.

Correct all errors before attempting to use the object again.

DSNU562I *csect name* **- TABLESPACE** *database.tablespace* **IS IN RECOVERY PENDING**

Explanation: The table space against which the utility was running is in recovery pending status. A RECOVER TABLESPACE is required.

Severity: 4 (warning)

System action: No action is taken.

User response: Take one of the following actions:

- Run the RECOVER TABLESPACE utility on the table space.
- Run the LOAD REPLACE utility on the table space.
- Use the REPAIR SET NORCVRPEND utility or the START DATABASE FORCE command to reset the recovery pending status. However, these two options will not correct the table space.

DSNU563I *csect name***-** *obj-type obj-qual.obj-name* **IS IN CHECK PENDING**

Explanation: This message is issued to report the check pending state of the specified object.

Severity: 4 (warning)

User response: If the object is a table space with referential relationships, then run the CHECK DATA utility to check for referential integrity violations.

If the object is a base table space with LOB columns, then run the CHECK DATA utility to validate the auxiliary columns.

If the object is a LOB table space, then run the CHECK LOB utility to validate the structure of the LOB table space.

If the object is an index space, then run the CHECK INDEX utility to verify that the index keys are consistent with the data in the table space or table space partition.

Correct all errors before attempting to use the object again.

DSNU564I *csect* **- RESTART NOT ALLOWED FOR** *obj-type obj-qual.obj-name*

Explanation: Restart of the RECOVER utility is not allowed for one of the following reasons:

1. During the RESTORE phase, access to image copies failed. Therefore, RECOVER determined an earlier point of recovery to search for available image copies. RECOVER might not be able to recreate the original failures for restart.
2. The utility uses the LOGONLY option and was previously restarted with RESTART(CURRENT). It cannot now be started with RESTART(PHASE).

Severity: 8 (error)

System action: The utility job terminates. If RECOVER acts on a list of objects, the restarted job terminates with the first object for which restart is not allowed.

User response: For case 1, execute REPORT RECOVERY on the object and also collect the output from the previous execution of RECOVER TABLESPACE. Using those two outputs and the console log, determine the reason for the failure during the RESTORE phase. Correct the reason for the failure, if necessary. Terminate the outstanding utility then rerun the RECOVER TABLESPACE without RESTART.

For case 2, restart the utility with RESTART(CURRENT) or restore the data sets and rerun the RECOVER job from the beginning.

Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1,

---

DSNU565I *csect-name* **CHECK CONSTRAINT** *constraint-name* **ON TABLE** *table-name* **WAS CREATED AFTER THE TARGET RECOVERY POINT**

Explanation: The check constraint *constraint-name* was created on table *table-name* after the target recovery point. The target recovery point is the specified time or RBA in the point-in-time recovery. The associated table space was placed in check pending status because the RECOVER utility cannot determine whether check integrity was maintained. Data in the table might have constraint violations.

Severity: 4 (warning)

System action: RECOVER processing continues.

User response: Refer to Part 2 (Volume 1) of *DB2 Administration Guide* for information about resolving the check pending status.

DSNU566I *csect-name* **RESTORE OF** *obj-type obj-qual.obj-name* **FROM DATA SET** *data-set-name* **COMPLETED, ELAPSED TIME =** *hh*:*mm*:*ss*

Explanation: This message is issued at the successful completion of the RESTORE phase of the RECOVER utility when the DFSMSdss RESTORE command is issued. The *data-set-name* identifies the DFSMS concurrent copy that was being restored. The elapsed time of the RESTORE is shown in hours, minutes, and seconds.

System action: RECOVER utility processing continues.

User response: No action is required.

DSNU567I *csect-name* **TABLESPACE=** *dbname.tsname* **PART NUMBER =** *partition number* **IS IN REORG PENDING STATE**

Explanation: A SYSCOPY record for a range of partitions has be encountered. The RECOVER TABLESPACE utility was executed against an object which was the target of a partition rebalancing. The RECOVER utility will set all of the partitions involved in the partition rebalancing to REORG PENDING for one of the following reasons:

- Recovery of a table space to a point in time prior to the resetting of the REORG PENDING state by the REORG utility.
- Recovery of a table space to currency, when the table space is in REORG PENDING state.

Severity: 4 (warning)

System action: The RECOVER utility completed successfully.

User response: Execute REORG with SHRLEVEL NONE on the partitions in REORG PENDING status to rebalance the partitions and to remove the REORG PENDING state.

DSNU568I *csect-name obj-type obj-qual.obj-name* **IS IN INFORMATIONAL COPY PENDING**

Explanation: An image copy of the index is required to establish a new base from which the index can be recovered. This is an advisory state, so the index is available for read or write access.

Severity: 0 (informational)

User response: Take an image copy of the index.

DSNU569I *csect-name* **RESTORE OF** *obj-type obj-qual.obj-name* **FROM DATA SET** *data-set-name* **FAILED WITH RETURN CODE** *return-code*

Explanation: This message is issued in the RESTORE phase of the RECOVER utility. The DFSMSdss RESTORE command was issued to restore the DFSMS concurrent copy named *data-set-name*. The return code is from DFSMSdss RESTORE.

Severity: 4 (warning)

System action: The RECOVER utility searches for the next recoverable point. DFSMSdss issues ADR messages to the job output to indicate the reason for the return code.

User response: You might want to investigate why this DFDSS RESTORE command failed. Collect any console messages. Look up the return code and any ADR messages in the appropriate DFSMSdss manual.

Problem determination: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 85,

DSNU570I *csect-name obj-type obj-qual.obj-name* **PARTITION** *part-num* **IS IN INFORMATIONAL COPY PENDING**

Explanation: An image copy of the index partition is required to establish a new base from which the index partition can be recovered. This is an advisory state, so the index is available for read or write access.

Severity: 0 (informational)

User response: Take an image copy of the index partition or of the entire index, depending on your backup and recovery procedures.

DSNU571I *csect-name* **ALL** *site* **SYSCOPY RECORDS FOR** *obj-type*= *obj-qual.obj-name* **PARTITION=***part-num* **HAVE BEEN DELETED**

Explanation: The MODIFY RECOVERY utility deleted all of the SYSCOPY table records for the specified table space or index space partition. Any related SYSLGRNX rows are deleted if DSNUM ALL is specified. The 'part-num' of specified table space or index space is no longer recoverable at the 'site'. The 'site' can be 'LOCALSITE' or 'RECOVERYSITE'.

Severity: 4 (warning)

System action: Utility processing continues.

System programmer response: The 'part-num' of specified table space or index space is no longer recoverable at the 'site' after this message is received. Use the COPY utility to create one or more copies for the 'site' to establish a base for future recovery.

DSNU572I *csect-name* **ALL** *site* **SYSCOPY RECORDS FOR** *obj-type*= *obj-qual.obj-name* **HAVE BEEN DELETED**

Explanation: The MODIFY RECOVERY utility deleted all of the SYSCOPY table records for the specified table space or index space. Any related SYSLGRNX rows are deleted if DSNUM ALL is specified. The table space or index space is no longer recoverable at the 'site'. The 'site' can be 'LOCALSITE' or 'RECOVERYSITE'.

Severity: 4 (warning)

System action: Utility processing continues.

System programmer response: The table space or index space is no longer recoverable at the 'site' after this message is received. Use the COPY utility to create one or more copies for the 'site' to establish a base for future recovery.

DSNU573I *csect-name* **NO BACKUP COPIES FOUND FOR THIS TABLESPACE OR DATASET**

Explanation: The MODIFY RECOVERY utility was unable to locate any SYSCOPY records older than the specified AGE or prior to the specified DATE. No records have been deleted.

Severity: 4 (warning)

System action: Processing completes normally.

DSNU575I *csect-name* **- MODIFY COMPLETED SUCCESSFULLY**

Explanation: This message reports the successful completion of the MODIFY utility. The requested cleanup of old SYSIBM.SYSCOPY entries was completed.

Severity: 0 (informational)

System action: The MODIFY utility completed normally.

DSNU577I *csect-name* **MODIFY HAS NOT DELETED ALL SYSCOPY RECORDS AS REQUESTED.**

Explanation: If MODIFY RECOVERY is invoked with a partition-number (DSNUM), and the table space level recoverable point is older than the user specified DATE or AGE, all SYSCOPY records after the table space level recoverable point cannot be deleted.

Severity: 4 (warning)

System action: Processing continues normally.

DSNU578I *csect-name* **SYSLGRNX INFORMATION FOR MEMBER** *member-name*

Explanation: In the data sharing environment, the SYSLGRNX records reported with DSNU513 are grouped by the respective member name. Each group member has header message DSNU578 to identify the member. This message precedes DSNU513.

System action: The RECOVER utility continues with the LOGAPPLY phase.

DSNU579I *csect-name* **RECOVER UTILITY LOG APPLY AT LOGPOINT** *xxx*

Explanation: This message shows the progress of the LOGAPPLY phase of the RECOVER utility. The highest LOGPOINT that was applied is *xxx*. The value of logpoint is RBA in a non data-sharing environment and LRSN in a data sharing environment.

System action: The LOGAPPLY phase continues normally.

DSNU580I *csect-name* **REPORT UTILITY COMPLETE, ELAPSED TIME=***hh/mm/ss*

Explanation: The message is issued at the normal completion of the REPORT utility. The time that elapsed since the beginning of the REPORT utility to its completion was 'hh' hours, 'mm' minutes, and 'ss' seconds.

Severity: 0 (informational)

System action: This message indicates the normal completion of the REPORT utility. Processing continues to the next utility statement in the utility job step, if any. If not, the utility job step is normally terminated, and the resources held become available for further processing. The UTILID of the utility job step becomes available for reuse.

DSNU581I *csect-name* **REPORT RECOVERY** *obj-type obj-qual.obj-name*

Explanation: This message indicates the beginning of the recovery information report for a table space or index space.

Severity: 0 (informational)

System action: Utility processing continues.

DSNU582I *csect-name* **REPORT RECOVERY** *obj-type obj-qual.obj-name* **SYSCOPY ROWS**

Explanation: This is the header message for the SYSCOPY section of the REPORT RECOVER output. This section displays the SYSCOPY rows for the table space or index space. Whether the information is displayed at the partition level is determined by the DSNUM specification in the REPORT RECOVERY input statement. The fields in the SYSCOPY row are:

- TIMESTAMP
- ICTYPE

  The following delimiters might be present around the ICTYPE value:

  – * * - Non image copy syscopy entry

  – < > - Image copy prior to rebalancing of table space partitions

  – ( ) - Image copy prior to LOG(NO) event. For image copies of indexes, the LOG(NO) event may have occurred on its underlying table space.
- DSNUM
- FILESEQNO
- START_RBA
- SHRLEVEL
- DEVTYPE
- ICBACKUP
- DSNAME
- STYPE
- PIT_RBA
- DSVOLSER
- HIGHDSNUM
- LOWDSNUM

For description of the fields, see SYSIBM.SYSCOPY table in Appendix D of *DB2 SQL Reference*.

Severity: 0 (informational)

DSNU583I *csect-name* **SYSLGRNX ROWS FROM REPORT RECOVERY FOR** *obj-type obj-qual.obj-name*

Explanation: This is the header message for the SYSLGRNX section of the REPORT RECOVERY output. This section displays the SYSLGRNX rows for the table space, index space, or partition as specified in the REPORT RECOVERY input statement. The fields in the SYSLGRNX row are:

- UCDATE - Modification date
- UCTIME - Modification time
- START RBA - Starting RBA
- STOP RBA - Stopping RBA
- START LRSN - Starting LRSN
- STOP LRSN - Stopping LRSN
- DSNUM - Partition number
- MEMBER - Data sharing member

An invalid SYSLGRNX record is marked with '**' at the end. The invalid SYSLGRNX record cannot be used for recovery because the range is outside recovery. If records are marked with '**', then a DSNU594 message should have been issued prior to this.

A SYSLGRNX record created after the last FULL IMAGE COPY, LOAD REPLACE LOG YES, or REORG LOG YES is marked with '*' at the end.

Severity: 0 (informational)

System action: Processing continues normally.

DSNU584I *csect-name* **REPORT RECOVERY** *obj-type obj-qual.obj-name* **BSDS VOLUMES**

Explanation: Message for the BSDS VOLUMES section of the REPORT RECOVERY output. This section contains BSDS volumes for the archive log (1 or 2) records associated with the table space or index space specified in the REPORT RECOVERY input statement. The output contains the following information:

```
START TIME END TIME START RBA  END RBA UNIT VOLSER DATA SET NAME
start      end      rba       rba  device volume data set name
timestamp timestamp
```

An archive log record created after the last FULL IMAGE COPY, LOAD REPLACE LOG YES, or REORG LOG YES is marked with '*' at the end.

System action: Processing continues normally

DSNU585I *csect-name* **REPORT RECOVERY** *obj-type obj-qual.obj-name* **CURRENT**

Explanation: The message is issued in response to the CURRENT option of REPORT RECOVERY utility. It indicates that only the information since the last

recoverable point for the specified table space or index space is reported.

Severity: 0 (informational)

System action: Normal utility processing continues.

DSNU586I *csect-name* **REPORT RECOVERY** *obj-type obj-qual.obj-name* **SUMMARY:**

Explanation: This is the header message for the SUMMARY section of the REPORT RECOVERY utility. All the volumes to be used by the RECOVER utility are summarized.

The volume name is marked with '*' at the end if any archive log record in the volume was created after the last FULL IMAGE COPY, LOAD REPLACE LOG YES, or REORG LOG YES.

System action: Processing continues normally

DSNU587I *csect-name* **REPORT TABLESPACE SET WITH TABLESPACE** *dbname.tsname*

Explanation: Header message for the REPORT TABLESPACESET output. The output contains all table spaces, tables, and dependent tables associated with the table space specified in the REPORT TABLESPACESET input statement.

Severity: 0 (informational)

System action: Utility processing continues normally.

DSNU588I *csect-name* **NO DATA TO BE REPORTED**

Explanation: The message is issued when no data can be found for a section of the REPORT output.

Severity: 0 (informational)

System action: Utility processing continues.

DSNU589I *csect-name* **REPORT RECOVERY** *obj-type obj-qual.obj-name* **COMPLETE**

Explanation: The message is issued at the normal completion of reporting the RECOVERY information for the table space or index space by the REPORT RECOVERY utility.

Severity: 0 (informational)

System action: This message indicates the normal completion of the REPORT RECOVERY utility. Processing continues to the next utility statement in the utility job step, if any. If not, the utility job step is normally terminated, and the resources held become available for further processing. The UTILID of the utility job step becomes available for reuse. The output from REPORT RECOVERY is now available for RECOVER utility use.

DSNU590I *csect-name object* **RESOURCE NOT AVAILABLE -** *object-name* **REASON = X'***reason-code***'**

Explanation: A required object *object-name* could not be accessed for the reason shown in *reason-code*. The required object can be a database, table space, or index space. If the object is in a stopped state and has partitions, then the stopped state applies to one or more of the partitions.

Severity: 8 (error) or 4 (warning)

System action: If the return code is 8, utility processing is terminated. If the return code is 4, utility processing continues to the next request.

User response: Determine the cause for unavailability. The -DISPLAY DATABASE command can be used to determine the status. After ensuring the object is available, resubmit the job.

DSNU591I *csectname* **AN UNSUCCESSFUL COMPLETION OF A LOG READ, REASON=X'***reason-code***'**

Explanation: The log manager subcomponent of DB2 detected an error during a LOG READ of SYSCOPY record for DSNDB06.SYSCOPY, DSNDB01.DBD01 or DSNDB01.SYSUTIL.

Severity: 4 (warning)

System action: Processing continues.

Problem determination: Look up the reason code in *DB2 Codes* for the problem determination.

DSNU592I *csect-name* **REPORT RECOVERY INFORMATION FOR DATA SHARING MEMBER** *member-name*

Explanation: The REPORT RECOVERY utility displays this message as the heading for SYSLGRNX and BSDS records or BSDS summary information for each data sharing member. Records for this member follow, if there is data to report.

Severity: 0 (Informational)

System action: Utility processing continues.

DSNU593I *csect-name* **REPORT RECOVERY ENVIRONMENT INFORMATION: MINIMUM RBA:** *rba1* **MAXIMUM RBA:** *rba2* **MIGRATING RBA:** *rba3*

Explanation: The REPORT utility issues this message to display recovery environment information used by the RECOVER utility. The maximum and minimum RBAs give the range of possible recovery points on the log before the start of data sharing. The migrating RBA is the RBA at the time of migration from DB2 Version 3 to Version 4. If there is data to report, other messages

from the REPORT utility follow to display SYSCOPY, SYSLGRNX and BSDS records.

Severity: 0 (Informational)

System action: Utility processing continues.

DSNU594I *csect-name* **DB2 ENCOUNTERED SYSLGRNX RECORDS THAT CANNOT BE USED FOR RECOVERY.**

Explanation: The utility found one or more SYSLGRNX records that cannot be used for recovery because the range is outside recovery. Invalid SYSLGRNX records are marked with '**' at the end of the record under message DSNU583I.

Severity: 4 (REPORT - warning) or 8 (RECOVER - error)

System action: REPORT utility processing continues. RECOVER utility terminates.

System programmer response: Run the RECOVER utility with the TOCOPY option.

DSNU595I *csect-name* **ERROR IN GETTING THE HIGHEST MEMBER ID IN DATA SHARING.**

Explanation: The REPORT utility could not find a valid highest member ID. The highest member is the last member to join a data sharing group. This message is issued only in a data sharing environment.

Severity: 8 (error)

System action: The REPORT utility abends.

System programmer response: This might be an internal DB2 problem. Refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNU596I *csect-name* **ERROR IN GETTING THE MEMBER NAME FOR MEMBER ID**

Explanation: The REPORT utility could not find a valid member name for the member ID, or the member ID supplied is invalid. This message is issued only in a data sharing environment.

Severity: 8 (error)

System action: The REPORT utility abends.

System programmer response: This might be a DB2 internal problem. Refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNU597I *csect-name* **INVALID SYSLGRNX INFORMATION FROM REPORT RECOVERY**

Explanation: This is the header message for the invalid SYSLGRNX records section of the REPORT RECOVERY output. This section lists SYSLGRNX records containing information that is invalid for recovery purposes.

Severity: 0 (Informational)

System action: Utility processing continues.

User response: Examine the invalid SYSLGRNX records with a zero member ID which were generated before enabling data sharing. The MODIFY utility can be used to delete the invalid SYSLGRNX records.

DSNU599I *csect-name obj-type obj-qual.obj-name* **HAS BEEN RECOVERED TO A CONSISTENT POINT IN TIME WITH** *obj-type obj-qual.obj-name*

Explanation: This message is issued when an index and its underlying table space have been recovered to a consistent point in time. A consistent point in time is a common QUIESCE point or a common SHRLEVEL REFERENCE copy set (objects were copied in the same list with SHRLEVEL REFERENCE specified). The index is available for read and write access when the recover completes.

Severity: 0 (informational)

System action: Recover processing continues.

DSNU600I *csect-name* **INDEX** *creator.index-name* **POINTS TO A DIFFERENT DATABASE.**

Explanation: A specified index does not point to a table in the database selected. For a RUNSTATS Format 1 invocation, the database is the one specified (or defaulted) in the TABLESPACE parameter. For a RUNSTATS Format 2 invocation, the database is the one which contains the first index in the index-name list.

Severity: 8 (error)

System action: All remaining RUNSTATS invocation parameters are validated and the executing utility job terminates. Work performed by previous utilities in the job step remains in effect. Subsequent utility control statements are flushed. No record of the utility job step is retained in the system; therefore, it cannot be restarted.

User response: Ensure that all indexes in the list point to the same database and resubmit the RUNSTATS job.

Problem determination: Refer to catalog table SYSIBM.SYSINDEXES to determine the table upon which the index is based. This table must be in the

database and table space targeted by the RUNSTATS utility.

DSNU601I *csect-name* **INDEX** *creator.index-name* **POINTS TO A DIFFERENT TABLESPACE.**

Explanation: A specified index does not point to a table in the table space selected. The table space is either the one specified in the TABLESPACE parameter or the one which contains the table pointed to by the first specified index.

Severity: 8 (error)

System action: All remaining RUNSTATS invocation parameters are validated and the executing utility job terminates. Work performed by previous utilities in the job step remains in effect. Subsequent utility control statements are flushed. No record of the utility job step is retained in the system; therefore, it cannot be restarted.

User response: Ensure that all indexes in the list point to tables in the targeted table space and resubmit the RUNSTATS job.

Problem determination: Refer to catalog table SYSIBM.SYSINDEXES to determine the table upon which the index is based. Using the table-name found in SYSIBM.SYSINDEXES, refer to catalog table SYSIBM.SYSTABLES to determine the containing table space.

DSNU602I *csect-name* **STATISTICS ARE NOT COLLECTED FOR NONPARTITIONED INDEX** *creator.index-name*

Explanation: For a LOAD or REORG utility when a nonpartitioned index exists, inline statistics performs a complete index scan. Statistics are collected for the entire nonpartitioned index. This warning message will be issued if the PART keyword & STATISTICS keyword are specified for LOAD/REORG utilities. This is because statistics are not collected on a logical part basis.

System action: This message will be issued for each nonpartitioned index, when INDEX ALL is specified.

Severity: 4 (warning)

User response: Run the runstats utility on the object.

DSNU603I *csect-name* **- OBJECT=***object-name* **HAS AN INCOMPLETE DEFINITION**

Explanation: The identified object (TABLE or TABLE SPACE) does not have a complete definition. This can be caused by one of these errors:

- Not defining any tables in the table space (RC=4).
- Not creating the accompanying partitioned index if the table space is partitioned (RC=8).

Severity: 8 (error) or 4 (warning)

System action: If the return code is 8, utility processing terminates. If the return code is 4, utility processing continues to the next request.

User response: Complete the table space definition by creating at least one table in the table space, or create the accompanying partitioned index if the table space is partitioned.

Problem determination: Issue an SQL SELECT statement on the catalog to determine if the tables or a partitioned index was not created for the table space.

DSNU604I *csect-name object* **IS EMPTY**

Explanation: The object currently to be scanned by the RUNSTATS utility contains no records. The 'object' can be TABLESPACE or INDEXSPACE.

Severity: 0 (informational)

System action: The object currently to be scanned is recorded in the catalog as empty. The statistics for the object are updated and RUNSTATS continues to the next object, if there is one.

DSNU605I *csect-name* **ERROR OCCURRED IN ACCESSING TABLESPACE** *database-name.tablespace-name*

Explanation: A previous error makes it impossible for RUNSTATS to scan TABLESPACE 'creator.tbspname'.

Severity: 8 (error)

System action: The executing utility job step terminates. Work performed by previous utilities in the job step remains in effect. Subsequent utility control statements are flushed. No record of the utility job step is retained in the system; therefore, it cannot be restarted.

User response: Examine prior error messages for error identification. If the error can be corrected, resubmit the RUNSTATS utility in a new utility job step. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: Prior messages identify this error. See the information about these messages for additional problem determination.

DSNU606I *csect-name* **USER NOT AUTHORIZED TO ACCESS TABLESPACE** *database-name.tablespace-name*

Explanation: This user is not authorized to execute the RUNSTATS utility against TABLESPACE 'creator.tbspname'.

Severity: 8 (error)

System action: The executing utility job step

terminates. Work performed by previous utilities in the job step remains in effect. Subsequent utility control statements are flushed. No record of the utility job step is retained in the system; therefore, it cannot be restarted.

User response: Contact your system administrator to obtain the proper authority, or have another user with the required authority execute the utility for you.

DSNU607I *csect-name* **ERROR OCCURRED IN ACCESSING INDEX** *creator.index-name*

Explanation: A previous error makes it impossible for RUNSTATS to scan INDEXSPACE 'creator.indexspacename'.

Severity: 8 (error)

System action: The executing utility job step terminates. Work performed by previous utilities in the job step remains in effect. Subsequent utility control statements are flushed. No record of the utility job step is retained in the system; therefore, it cannot be restarted.

User response: Examine prior error messages for error identification. If the error can be corrected, resubmit the RUNSTATS utility in a new utility job step.

Problem determination: Prior messages identify this error. See the documentation for these messages in this manual for additional problem determination. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNU608I *csect-name* **USER NOT AUTHORIZED TO ACCESS INDEXSPACE** *creator.index-name*

Explanation: This user is not authorized to execute the RUNSTATS utility against INDEXSPACE 'creator.indexspacename'.

Severity: 8 (error)

System action: The executing utility job step terminates. Work performed by previous utilities in the job step remains in effect. Subsequent utility control statements are flushed. No record of the utility job step is retained in the system; therefore, it cannot be restarted.

User response: Contact your system administrator to obtain the proper authority, or have another user with the required authority execute the utility for you.

DSNU609I *csect-name* **TABLE** *table-name* **IS IN A DIFFERENT TABLESPACE**

Explanation: A table listed in the table name list of the RUNSTATS TABLE keyword was not found in the specified table space.

Severity: 8 (error)

System action: The executing utility job step terminates. Work performed by previous utilities in the job step remains in effect. Subsequent utility control statements are flushed. No record of the utility job step is retained by the system; therefore it cannot be restarted.

User response: Ensure that all tables in the list are in the specified table space and resubmit the utility.

Problem determination: Refer to the catalog table SYSIBM.SYSTABLES to determine the table space in which the table is located.

DSNU610I *csect-name catalog-table-name* **CATALOG UPDATE FOR** *object-name* **SUCCESSFUL**

Explanation: This message indicates that the updating of the statistics columns in the CATALOG TABLE 'catalog-table-name' was successful.

Severity: 0 (informational)

System action: Processing continues with the next catalog table.

DSNU611I *csect-name* **COLUMN** *column-name* **NOT FOUND FOR TABLE** *creator.table-name*

Explanation: A column name, given as an operand of the COLUMN keyword, was not found in the specified table.

Severity: 8 (error)

System action: Validation of the utility invocation statement is completed and the executing utility job step is terminated.

User response: Ensure that the column name given in the message is correct for the table specified.

Problem determination: Check that the column name specified in the utility invocation statement is identical to that found in the SYSIBM.SYSCOLUMNS catalog table. Correct any errors and reissue the utility job.

DSNU612I *csect-name* **SYSTABLESPACE CATALOG STATISTICS FOR** *tspace-name* **NACTIVE =** *nactive* **NACTIVEF =** *nactive*

Explanation: The RUNSTATS utility issues this message to report the statistics collected for table space *tspace-name*.

nactive The number of active pages in the table space

This message is routed to SYSPRINT.

Severity: 0 (informational)

System action: RUNSTATS processing continues.

User response: Refer to *DB2 Utility Guide and Reference* for information about RUNSTATS statistics.

DSNU613I *csect-name* **SYSTABLEPART CATALOG STATISTICS FOR** *table-space-name* **PARTITION** *part-num*

CARD	= cardinality
CARDF	= cardinality
NEARINDREF	= nearindref
FARINDREF	= farindref
PERCACTIVE	= peractive
PERCDROP	= percdrop
PAGESAVE	= pagesave
SPACE	= space
SPACEF	= space
PQTY	= pqty
SQTY	= sqty
DSNUM	= dsnum
EXTENTS	= extents

Explanation: The RUNSTATS utility issues this message to report the statistics collected. If the *part-num* equals 0, the statistics apply to the entire table space. If the *part-num* is not 0, the statistics apply to the specified partition of the table space.

cardinality
The number of rows in the reported partition number

nearindref
The number of rows that were relocated near their original page

farindref
The number of rows that were relocated far from their original page

peractive
The percentage of space occupied by actual data from active tables

percdrop
The percentage of space occupied by rows of data from dropped tables

pagesave
The percentage of pages saved in the table space or partition by using data compression

space The space currently allocated for all extents, measured in kilobytes.

pqty The primary space allocation for the dataset in 4K blocks.

sqty The secondary space allocation for the dataset in 4K blocks.

dsnum The number of data sets.

extents The number of extents for the last data set of the object.

This message is routed to SYSPRINT.

Severity: 0 (informational)

System action: RUNSTATS processing continues.

User response: Refer to *DB2 Utility Guide and Reference* for information about RUNSTATS statistics.

DSNU614I *csect-name* **SYSTABLES CATALOG STATISTICS FOR** *table-name*

CARD	= cardinality
CARDF	= cardinality
NPAGES	= npages
NPAGESF	= npages
PCTPAGES	= pctpages
PCTROWCOMP	= pctrowcomp
AVGROWLEN	= avgrowlen
SPACEF	= space

Explanation: The RUNSTATS utility issues this message to report the statistics collected for the specified table. The statistics follow:

cardinality
The number of rows in the table.

For a LARGE table space, the value of CARD is set to -1.

npages The number of pages on which rows of the table appear.

pctpages
The percentage of total pages of the table space that contain rows from this table.

pctrowcomp
The percentage of rows compressed within the total number of active rows in the table.

avgrowlen
The total number of bytes divided by the number of rows.

space The space currently allocated for all extents, measured in kilobytes.

RUNSTATS uses floating point fields to report these statistics.

This message is routed to SYSPRINT.

Severity: 0 (informational)

System action: RUNSTATS processing continues.

User response: Refer to *DB2 Utility Guide and Reference* for a discussion of RUNSTATS statistics.

DSNU615I *csect-name* **SYSCOLUMNS CATALOG STATISTICS FOR** *column-name*

COLCARD	= cardinality
COLCARDF	= cardinality
HIGH2KEY	= high2key
LOW2KEY	= low2key

Explanation: The RUNSTATS utility issues this message to report the statistics collected for column *column-name*. Message DSNU619I is issued to identify the table.

cardinality
The number of distinct values in the column

high2key
The second highest value of the column

low2key
The second lowest value of the column

If the values for HIGH2KEY and LOW2KEY are null, the invoker of RUNSTATS is not authorized to read the table.

For a LARGE table space, the value of COLCARD is set to -1.

RUNSTATS uses floating point fields to report these statistics.

This message is routed to SYSPRINT.

Severity: 0 (informational)

System action: Processing continues.

User response: Refer to *DB2 Utility Guide and Reference* for a description of RUNSTATS statistics.

DSNU616I *csect-name* **SYSCOLDIST CATALOG STATISTICS FOR** *column-name*, *column-name*, *column-name* **...**

CARDINALITY = cardinality

FREQUENCY	COLVALUE
frequency	X “hex_string”
frequency	X “hex_string”
...	...
frequency	X “hex_string”

Explanation: The RUNSTATS utility issues this message to report the statistics collected for the indexed column 'column-name'. If more than one column is listed, then the statistics collected and the column values reported represent the concatenation of those columns.

The value for CARDINALITY is the number of distinct values for the listed columns. This statistic appears only if the KEYCARD option was specified.

The values under FREQUENCY are the same as those that are inserted into the DB2 catalog. Consider each 'frequency' as a four-digit number with an implied decimal point two places to the left of the final digit. The resulting number is the percentage of total rows in the table that contain a particular value for the named index columns.

Frequency is associated with the occurrence of a single value which is listed under COLVALUE.

Frequencies for the most frequent values are collected by default, or if the FREQVAL option is specified and given a COUNT value greater than zero. If the default is taken, then the ten most frequently occurring values are reported in descending order of frequency.

If fewer than ten values appear, then there are fewer than ten distinct values in the index.

If no values appear, then the values are uniformly distributed.

If frequencies appear but column values do not, then the invoker of RUNSTATS is not authorized to read the table. Message DSNU619I is issued to identify the table.

This message is routed to SYSPRINT.

Severity: 0 (informational)

System action: Processing continues.

User response: Refer to *DB2 Utility Guide and Reference* for a description of RUNSTATS statistics.

DSNU617I *csect-name* **SYSINDEXES CATALOG STATISTICS FOR** *index-name*

CLUSTERED	= clustered
CLUSTERRATIO	= clusterratio
CLUSTERRATIOF	= clusterratio
FIRSTKEYCARD	= firstkeycard
FIRSTKEYCARDF	= firstkeycard
NLEAF	= nleaf
NLEVELS	= nlevels
SPACEF	= space

Explanation: The RUNSTATS utility issues this message to report the statistics collected for index 'index-name'.

clustered
Whether the table is actually clustered by this index (YES or NO)

cluster-ratio
The percentage of data rows that are in clustering order for this index

float-cluster-ratio
A number between 0 and 1 that when multiplied by 100 gives the percentage of data rows in the partition that are in clustering order for this index

first-key
The number of distinct values of the first key column

full-key
The number of distinct values of the key

nleaf The number of active leaf pages in the index

nlevels The number of levels in the index tree

space The space currently allocated for all extents, measured in kilobytes.

For a LARGE table space, the values of FIRSTKEYCARD and FULLKEYCARD are set to -1. RUNSTATS uses floating point fields to report these statistics.

This message is routed to SYSPRINT.

Severity: 0 (informational)

System action: Processing continues.

User response: Refer to *DB2 Utility Guide and Reference* for a description of RUNSTATS statistics.

DSNU618I *csect-name* **SYSINDEXPART CATALOG STATISTICS FOR** *index-name* **PARTITION** *part-num*

CARD	=	cardinality
CARDF	=	cardinality
NEAROFFPOS	=	near-off
NEAROFFPOSF	=	near-off
FAROFFPOS	=	far-off
FAROFFPOSF	=	far-off
LEAFDIST	=	leafdist
SPACEF	=	space
DSNUM	=	dsnum
EXTENTS	=	extents
LEAFNEAR	=	leafnear
LEAFFAR	=	leaffar
PSEUDO_DEL_ENTRIES	=	pseudo_del_entries

Explanation: The RUNSTATS utility issues this message to report the statistics collected for index 'index-name'. If 'part-num' equals 0, the statistics apply to the entire index. If 'part-num' is not 0, the statistics apply to the specified partition of the index.

cardinality
The number of rows in the partition or index.

near-off
The number of rows pointed to that are near but not at their optimal positions.

far-off The number of rows pointed to that are far from their optimal positions.

leafdist
100 times the average distance in page IDs between successive leaf pages during a sequential access of the index.

space The space currently allocated for all extents, measured in kilobytes.

dsnum The number of data sets.

extents The number of extents for the last data set of the object.

leafnear
The number of leaf pages physically near the previous leaf page for successive active leaf pages.

leaffar The number of leaf pages located physically far away from the previous leaf pages for successive active leaf pages accessed in an index scan.

pseudo_del_entries
The number of entries that are logically deleted.

For a LARGE table space, the values of CARD, NEAROFFPOS, and FAROFFPOS are set to -1. RUNSTATS uses floating point fields to report these statistics.

This message is routed to SYSPRINT.

Severity: 0 (informational)

System action: Processing continues.

User response: Refer to *DB2 Utility Guide and Reference* for a description of RUNSTATS statistics.

DSNU619I *csect-name* **- USER** *auth_id* **DOES NOT HAVE PRIVILEGE ON TABLE** *table_name*

Explanation: The RUNSTATS utility issues this message to report that authorization ID 'auth_id' does not have privilege on table 'table_name'. Some column values reported by messages DSNU615I, DSNU616I, DSNU625I, and DSNU626I contain the null value.

Severity: 0 (informational)

System action: RUNSTATS processing continues.

User response: To obtain the key values, the user must be granted authority on the table. The RUNSTATS utility can be rerun with the report option.

DSNU620I *csect-name* **RUNSTATS CATALOG STATISTICS STATSTIME =** *timestamp*

Explanation: The RUNSTATS utility issues this message to report the timestamp value when statistics were collected. The value 'timestamp' is common to all of the statistic tables that were updated.

Severity: 0 (informational)

System action: Processing continues.

User response: Refer to *DB2 Utility Guide and Reference* for a description of the RUNSTATS statistics.

DSNU621I *csect-name* **RUNSTATS COMMIT/ROLLBACK FAILED**

Explanation: The RUNSTATS utility updated the catalog with gathered statistics and has tried either to commit those changes or to perform rollback processing to remove them following an error. The commit or rollback failed.

Severity: 8 (error)

System action: RUNSTATS processing is terminated. Subsequent utility control statements in the utility job step are executed.

System programmer response: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: This is an internal error.

DSNU622I *csect-name* **CATALOG ACCESS FOR** *object-name* **NOT SUCCESSFUL, REASON=***reason-code*

Explanation: Due to catalog access error for object *object-name* and reason code *reason-code*, the catalog update, insert, or select was not successful for the object when inline statistics was employed.

Severity: 4 (warning)

User response: Run the RUNSTATS utility against the object.

DSNU623I *csect-name* **-** *catalog-table-name* **CATALOG NOT UPDATED WITH AGGREGATE STATISTICS FOR** *object-name* **BECAUSE SOME PARTITIONS HAVE NO VALID STATISTICS**

Explanation: Aggregate statistics were requested, but partition statistics do not exist for every partition. Without partition statistics, the aggregate statistics cannot be calculated. The statistics for 'object-name' are not updated in the specified DB2 catalog table.

One possible cause for this message is that someone altered the length of a varying-length column and then tried to run RUNSTATS with the PART option.

Severity: 4 (warning)

System action: Processing continues.

User response: Run the RUNSTATS utility to collect statistics for:
- The entire table space, table, column or index, or
- Every partition.

DSNU624I *csect-name* **SYSTABSTATS CATALOG STATISTICS FOR** *table-name* **PARTITION** *part-num*

CARD or CARDF = cardinality
NPAGES = npages
PCTPAGES = pctpages
NACTIVE = nactive
PCTROWCOMP = pctrowcomp

Explanation: The RUNSTATS utility issues this message to report the statistics collected for the specified partition of the table space containing the table. The statistics follow:

cardinality
The number of rows in the table space partition

npages The number of pages on which rows of the table space partition appear

pctpages
The percentage of total pages of the table space partition that contain rows from this table

nactive The number of active pages in the table space partition

pctrowcomp
The percentage of rows compressed within the total number of active rows in the partition

This message is routed to SYSPRINT.

Severity: 0 (informational)

System action: RUNSTATS processing continues.

User response: Refer to *DB2 Utility Guide and Reference* for information about RUNSTATS statistics.

DSNU625I *csect-name* **SYSCOLSTATS CATALOG STATISTICS FOR** *column-name* **PARTITION** *part-num*

COLCARD = cardinality
HIGHKEY = highkey
HIGH2KEY = high2key
LOW2KEY = low2key
LOWKEY = lowkey

Explanation: The RUNSTATS utility issues this message to report the statistics collected for column 'column-name' partition 'part-num' of a table. Message DSNU619I is issued to identify the table.

cardinality
The number of distinct values in the column for the partition

highkey
The highest value of the column in the partition

high2key
The second highest value of the column in the partition

low2key
The second lowest value of the column in the partition

lowkey The lowest value of the column in the partition

If the values for 'highkey', 'high2key', 'low2key', and 'lowkey' are null, the invoker of RUNSTATS is not authorized to read the table.

This message is routed to SYSPRINT.

Severity: 0 (informational)

System action: Processing continues.

User response: Refer to *DB2 Utility Guide and Reference* for a description of RUNSTATS statistics.

DSNU626I *csect-name* **SYSCOLDISTSTATS CATALOG STATISTICS FOR** *column-name*, *column-name*, *column-name* **...** **PARTITION** *part-num*

CARDINALITY = cardinality

FREQUENCY	COLVALUE
frequency	X “hex_string”
frequency	X “hex_string”
...	...
frequency	X “hex_string”

Explanation: The RUNSTATS utility issues this message to report the statistics collected for the indexed column 'column-name' partition 'part-num'. If more than one column is listed, then the statistics collected and the column values reported represent the concatenation of those columns.

The value for CARDINALITY is the number of distinct values for the listed columns. This statistic appears only if the KEYCARD option was specified.

The values under FREQUENCY are the same as those that are inserted into the DB2 catalog. Consider each 'frequency' as a four-digit number with an implied decimal point two places to the left of the final digit. The resulting number is the percentage of total rows in the table that contain a particular value for the named index columns.

Frequency is associated with the occurrence of a single value which is listed under COLVALUE.

Frequencies for the most frequent values are collected by default, or if the FREQVAL option is specified and given a COUNT value greater than zero. If the default is taken, then the ten most frequently occurring values, plus the lowest and highest values in the partition (if not already included in the ten greatest frequencies), are reported in descending order of frequency.

If fewer than ten values appear, then there are fewer than ten distinct values in the index.

If no values appear, then the values are uniformly distributed.

If frequencies appear but column values do not, then the invoker of RUNSTATS is not authorized to read the table. Message DSNU619I is issued to identify the table.

This message is routed to SYSPRINT.

Severity: 0 (informational)

System action: Processing continues.

User response: Refer to *DB2 Utility Guide and Reference* for a description of RUNSTATS statistics.

DSNU627I *csect-name* **SYSINDEXSTATS CATALOG STATISTICS FOR** *index-name* **PARTITION** *part-num*

CLUSTERRATIO or CLUSTERRATIOF = cluster-ratio
FIRSTKEYCARD or FIRSTKEYCARDF = first-key
FULLKEYCARD or FULLKEYCARDF = full-key
NLEAF = nleaf
NLEVELS = nlevels
KEYCOUNT or KEYCOUNTF = keycount

Explanation: The RUNSTATS utility issues this message to report the statistics collected for partition *part-num* of index *index-name*.

cluster-ratio
The percentage of data rows in the partition that are in clustering order for this index

first-key
The number of distinct values of the first key column in the partition

full-key
The number of distinct values of the key in the partition

nleaf The number of active leaf pages in the index partition

nlevels The number of levels in the index tree in the partition

keycount
The number of index keys (entries) in the partition

This message is routed to SYSPRINT.

Severity: 0 (informational)

System action: Processing continues.

User response: Refer to *DB2 Utility Guide and Reference*

for a description of RUNSTATS statistics.

DSNU628I *csect-name* **STATISTICS WILL NO LONGER BE COLLECTED ON RESTART**

Explanation: Since inline statistics is not restartable, statistics might not be collected on the object that was being processed during the restart of the LOAD/REORG utility. If a list of objects is being processed, then objects that have not processed yet, will still have statistics collected. The error message will be issued on restart of a LOAD/REORG utility with STATISTICS keyword specified.

Severity: 4 (warning)

User response: Run the runstats utility on the object after the utility execution unless the DSNU620I message is present, indicating that the statistics were collected.

DSNU629I *csect-name* **INLINE STATISTICS IS NOT SUPPORTED FOR** *object-name*

Explanation: The collection of inline statistics is not supported on the following directory and catalog tablespaces : DSNDB06.SYSDBASE, DSNDB06.SYSDBAUT, DSNDB06.SYSGROUP, DSNDB06.SYSPLAN, DSNDB06.SYSVIEWS, DSNDB06.SYSSTATS, and DSNDB01.DBD01

This error message will be issued on execution of a REORG or REBUILD INDEX utility on a catalog object when the STATISTICS keyword is specified.

Severity: 4 (warning)

User response: Remove the keyword STATISTICS from the utility job statement and rerun the utility. Run the RUNSTATS utility afterwards.

DSNU630I *csect-name* **- SYSLOBSTATS CATALOG STATISTICS FOR** *LOB table space* **AVGSIZE =** *avgsize* **FREESPACE =** *freespace* **ORGRATIO =** *orgratio*

Explanation: The RUNSTATS utility issues this message to report statistics collected for the LOB table space.

AVGSIZE
Average size of a LOB in the LOB table space.

FREESPACE
Amount of free space in the LOB table space.

ORGRATIO
The percentage of organization in the LOB table space. The following values indicate the state of the LOB table space:

100.00 The table space is completely organized or it is empty.

1 The table space is disorganized.

0.00 The table space is completely disorganized.

System action: RUNSTATS processing continues.

DSNU631I *csect-name* **-** *catalog-table-name* **CATALOG NOT UPDATED WITH AGGREGATE STATISTICS FOR** *object-name* **BECAUSE ERROR FROM SORT COMPONENT RC=***n*

Explanation: The aggregate statistics cannot be calculated because of an error from the SORT routine. An MVS sort returned a non-zero return code *n*.

The statistics for *object-name* are not updated in the DB2 catalog table that was specified.

Severity: 4 (warning)

System action: Processing continues, but the statistics for *object-name* are not updated in the DB2 catalog table that was specified.

User response: Determine the cause of the error from the SORT routine. Run the RUNSTATS utility to collect statistics for:

- The entire tablespace, table, column, or index, or
- Every partition

DSNU635I **COLGROUP STARTING WITH** *first-column-name* **IS TOO LONG**

Explanation: The length of the COLGROUP value has exceeded the length of the COLVALUE column defined in the SYSIBM.SYSCOLDIST catalog table.

Severity: 8 (error)

System action: The executing utility job step terminates.

User response: Alter the COLGROUP specification so that the maximum length of the COLGROUP will not exceed the length of the COLVALUE column defined in the SYSIBM.SYSCOLDIST catalog table and resubmit the RUNSTATS utility. If the length of a COLGROUP with a single column specified exceeds the maximum length of COLVALUE, then the distribution statistics cannot be collected on that COLGROUP

DSNU638I *csect* **- UTILITY NOT ALLOWED TO RUN ON TABLESPACE =** *ttttttt* **IN ROSHARE READ DATABASE =** *dddddddd*

Explanation: An attempt to execute STOSPACE utility on a table space defined in a ROSHARE READ database failed. The STOSPACE utility is only allowed to execute in a ROSHARE OWNER or a nonshared database.

Severity: 4 (warning)

System action: Processing continues to the next table space or index space in the storage group, if any.

User response: Resubmit the utility job on the system in which the database was defined with the ROSHARE OWNER attribute.

DSNU639I *csect* **- UTILITY NOT ALLOWED TO RUN ON INDEX SPACE =** *iiiiiiii* **IN ROSHARE READ DATABASE =** *dddddddd*

Explanation: An attempt to execute the STOSPACE utility on an index space defined in a ROSHARE READ database failed. The STOSPACE utility is only allowed to execute in a ROSHARE OWNER or nonshared database.

Severity: 4 (warning)

System action: Processing continues to the next table space or index space in the storage group, if any.

User response: Resubmit the utility job on the system in which the database was defined with the ROSHARE OWNER attribute.

DSNU640I *csect-name* **DATA SET=** *nnn* **OF TABLESPACE=** *ttttttt* **IN DATABASE=** *dddddddd* **HAS BEEN MIGRATED BY HSM**

Explanation: The STOSPACE utility has found that a DB2 data set has been migrated by the Data Facility Hierarchical Storage Manager. The amount of storage space used cannot be determined for DATA SET 'nnn' of TABLESPACE 'ttttttt' in DATABASE 'dddddddd'. The space allocation information in the catalog will not include the storage used by the migrated data set.

Severity: 0 (informational)

System action: Processing continues to the next data set in the storage group, if any.

System programmer response: If required, the data set may be recalled using HSM, and the STOSPACE utility may be run again.

DSNU641I *csect-name* **DATA SET=** *nnn* **OF INDEXSPACE=** *iiiiiiii* **IN DATABASE=** *dddddddd* **HAS BEEN MIGRATED BY HSM**

Explanation: The STOSPACE utility has found that a DB2 data set has been migrated by the Data Facility Hierarchical Storage Manager. The amount of storage space used cannot be determined for DATA SET 'nnn' of INDEXSPACE 'iiiiiiii' in DATABASE 'dddddddd'. The space allocation information in the catalog will not include the storage used by the migrated data set.

Severity: 0 (informational)

System action: Processing continues to the next data set in the storage group, if any.

System programmer response: If required, the data set may be recalled using HSM, and the STOSPACE utility may be run again.

DSNU642I *csect-name* **UNABLE TO ACCESS DBD FOR DATABASE =** *dddddddd*, **TABLESPACE =** *ttttttt*

Explanation: A database descriptor control block could not be accessed by the STOSPACE utility. Information in the DBD is required before space allocation data can be obtained from the appropriate ICF catalog. Storage space used by table space 'ttttttt' in database 'dddddddd' cannot be determined.

Severity: 4 (warning)

System action: Processing continues with the next storage group.

User response: Determine the cause for unavailability. The database may be stopped or the DBD may be marked as invalid. The -DISPLAY DATABASE command can be used to determine the status.

System programmer response: Determine if the reason for the DBD unavailability may require system corrective action.

Problem determination: A DSNT501I message may have been sent to the operator console. If so, see the problem determination section for message DSNT501I in the Chapter 17, “DSNT... messages,” on page 283.

DSNU643I *csect-name* **UNABLE TO ACCESS DBD FOR DATABASE =** *dddddddd*, **INDEXSPACE =** *ttttttt*

Explanation: A database descriptor control block could not be accessed by the STOSPACE utility. Information in the DBD is required before space allocation data can be obtained from the appropriate ICF catalog. Storage space used by index space 'ttttttt' in database 'dddddddd' cannot be determined.

Severity: 4 (warning)

System action: Processing continues with the next storage group.

User response: Determine the cause for unavailability. The database may be stopped or the DBD may be marked as invalid. The -DISPLAY DATABASE command can be used to determine the status.

System programmer response: Determine if the reason for the DBD unavailability may require system corrective action.

Problem determination: A DSNT501I message may have been sent to the operator console. If so, see the problem determination section for message DSNT501I in the Chapter 17, “DSNT... messages,” on page 283 section.

DSNU644I *csect-name* **USER** *user-name* **NOT AUTHORIZED FOR STOSPACE UTILITY**

Explanation: Utility processing for the user indicated on the job statement failed, because the user is not authorized to execute the STOSPACE utility.

Severity: 8 (error)

System action: Utility processing is terminated without invoking the requested utility or utilities.

User response: Contact the systems programmer for assistance with your security classification.

System programmer response: If the USER specified on the job statement is incorrect, correct it, and resubmit the job. To run the STOSPACE utility, the user must either have SYSADM authority or have been granted the STOSPACE privilege. Use the SQL GRANT command to give the user the desired authority.

DSNU645I *csect-name* **NO STORAGE GROUPS NAME FOUND**

Explanation: You requested the STOSPACE utility to report on all storage groups, but no storage group was found.

Severity: 4 (warning)

This condition does not prevent subsequent processing.

System action: This message indicates completion of the STOSPACE utility. Processing continues to the next utility statement in the utility job step, if any. If not, the utility job step is normally terminated. Resources are released to their prior availability state. The UTILID of the utility job step becomes available for reuse.

User response: This message indicates that there are no storage groups defined. Consult the system programmer to determine the reason for this condition.

System programmer response: This message indicates either that no storage groups were defined by the installation process (or by subsequent SQL procedures) or they have all been dropped. If this is an error, the required storage groups must be re-created with the CREATE STOGROUP SQL statement.

DSNU646I *csect-name* **STORAGE GROUP NAME** *nnnnnnnn* **NOT FOUND**

Explanation: This message is issued by STOSPACE when one of the requested storage groups is not found.

The following circumstances may exist:

1. When an inconsistency occurs between SYSIBM.SYSSTOGROUP at the beginning of STOSPACE execution and SYSIBM.SYSSTOGROUP during or at the end of processing. The inconsistency may be due to a storage group being dropped during the processing of STOSPACE utility.
2. The storage group name is misspelled.

Severity: 4 (warning)

System action: Processing proceeds normally to the next storage group if any.

User response: Verify the existence or spelling of the storage group names by referring to SYSIBM.SYSSTOGROUP. Submit another request with the correct storage group designated.

DSNU647I *csect-name* **UNAVAILABLE REASON X** *reason-code* **UNAVAILABLE TYPE X** *type-code*, **TABLESPACE NAME=** *table-space-name* **NOT FOUND**

Explanation: The indicated table space was not available for processing by the STOSPACE utility. The reason for its unavailability is given in 'reason-code'. The type of unavailability is given in 'type-code'.

This message may be issued when an inconsistency between SYSIBM.SYSSTABLESPACE and SYSIBM.SYSTABLEPART has been detected. The inconsistency may be due to a table space being dropped during the processing of the STOSPACE utility.

Severity: 4 (warning)

System action: Processing continues to the next table space in the storage group, if any.

User response: Use the 'reason-code' and 'type-code' to determine why the resource was unavailable. When the problem has been resolved, resubmit the STOSPACE request.

System programmer response: Determine if the 'reason-code' and 'type-code' require system corrective action.

Problem determination: See the Problem Determination section for message DSNT500I on page 335.

DSNU648I *csect-name* **UNAVAILABLE REASON X** *reason code*, **UNAVAILABLE TYPE X** *type-code*, **INDEX NAME=***index-name* **NOT FOUND**

Explanation: The indicated index was not available for processing by the STOSPACE utility. The reason for its unavailability is given in 'reason-code'. The type of unavailability is given in 'type-code'.

This message may be issued when an inconsistency between SYSIBM.SYSINDEXES and SYSIBM.SYSINDEXPART has been detected. The inconsistency may be due to an index being dropped during the processing of the STOSPACE utility.

Severity: 4 (warning)

System action: Processing continues to the next index in the storage group, if any.

User response: Use the 'reason-code' and 'type-code' to determine why the resource was unavailable. When the problem has been resolved, resubmit the STOSPACE request.

System programmer response: Determine if the 'reason-code' and 'type-code' require system corrective action.

Problem determination: See the Problem Determination section for message DSNT500I on page 335.

DSNU650I *csect-name control-statement-text*

Explanation: Each control statement is returned to the invoker as a part of the normal output. If an error occurred while processing the statement, a separate error message follows this message.

Severity: 0 (informational)

System action: The processing of the utility request continues normally.

DSNU651I *csect-name* **SET NOCOPYPEND OPERATION SUCCESSFUL**

Explanation: This message reports the successful completion of the REPAIR SET TABLESPACE NOCOPYPEND utility operation. The COPY PENDING restriction has been removed from the specified table space.

Severity: 0 (informational)

System action: The REPAIR SET statement has completed normally.

DSNU652I *csect-name* **VERIFY OPERATION SUCCESSFUL**

Explanation: This message is issued during REPAIR utility processing in response to a VERIFY control statement. It indicates the verification requested was successful, permitting a subsequent REPLACE request to take effect.

Severity: 0 (informational)

System action: The requested verification was successful. If this is the only verification required for the subsequent REPLACE control statement, the REPLACE control statement takes effect. If there are other verification control statements, they must also complete successfully before the subsequent REPLACE control statement takes effect.

DSNU653I *csect-name* **REPLACE OPERATION AT OFFSET** *hex-data* **WITH DATA** *hex-data* **FAILED. ATTEMPTED TO REPLACE CRITICAL NON-DATA INFORMATION ON THE PAGE** *page-num* **WITH INCORRECT INFORMATION.**

Explanation: The REPAIR utility job with the LOG YES option was unable to replace certain critical, nondata information in the page for one of these reasons:

- Some critical information on any page at the indicated offset cannot be replaced by this method.
- The page is a header page and some of the information at the indicated offset cannot be replaced by this method.
- The information can be replaced only by using the LOG NO option.

Severity: 8 (error)

System action: REPAIR resumes executing at the next LOCATE statement. Later control statements in the same LOCATE block are not executed.

User response: Correct the REPAIR control statements and resubmit the request in a new utility job step.

User response: Check the OFFSET option on the REPLACE statement to be sure it does not cause critical information to be replaced. If the replacement is intentional, use the LOG NO option.

DSNU654I *csect-name* **VERIFY OPERATION FAILED, DATA IS** *hex-data*

Explanation: The verify operation requested on the control statement failed. The actual data is given as 'hex-data'.

Severity: 4 (warning)

System action: All REPAIR REPLACE actions subject to the current LOCATE scope are ignored. Processing continues at the next LOCATE request, if any.

User response: Determine why the data (as reported in the message) does not match the expectations of VERIFY option. Make the appropriate correction, and resubmit the REPAIR request in a new utility job step.

Problem determination: The REPAIR DUMP option can help you determine the correct verification data. Refer to *DB2 Utility Guide and Reference* for REPAIR processing constraints.

DSNU655I *csect-name* **DELETE OPERATION SUCCESSFUL**

Explanation: This message is issued during the REPAIR utility in response to a DELETE control statement.

Severity: 0 (informational)

System action: The data record specified on the previous LOCATE statement and all of its related INDEX entries have been deleted from the system. Processing continues.

DSNU656I *csect-name* **REPLACE OPERATION SUCCESSFUL, DATA WAS** *hex-data*

Explanation: This message is issued by the REPAIR utility in response to a REPLACE control statement request. The message indicates that replacement of the data occurred and that the replaced data had the indicated hexadecimal value ('hex-data').

Severity: 0 (informational)

System action: The REPAIR utility located the indicated record, performed all requested verifications, and replaced the data in the record at the indicated offset with the new data as defined by the REPLACE control statement.

System programmer response: If a changed record is recovered before an image copy is made, and the REPAIR utility was run with LOG NO, the repair must be reapplied. If the REPAIR utility was run with LOG YES, no image copy is required before recovering the data after the repair.

DSNU657I *csect-name* **REPLACE RESET OPERATION SUCCESSFUL**

Explanation: This message is issued by the REPAIR utility in response to a REPLACE RESET control statement request. The message indicates that the broken page indicator has been reset.

Severity: 0 (informational)

System action: The REPAIR utility has reset the broken page indicator for the specified page. Although the reset indicator has been committed, it has not been logged. If the page is recovered before an image copy is made, the REPAIR RESET must be reapplied.

However, to make the repair persist across subsequent RECOVER operations, an image copy should be taken subsequent to the completion of the REPAIR utility.

DSNU658I *csect-name* **MULTIPLE RECORDS FOUND WITH SPECIFIED KEY**

Explanation: The specified key is not unique. Several DSNU660I messages, giving possible RID values for this KEY, follow this message.

Severity: 8 (error)

System action: Processing continues normally.

User response: Change the LOCATE statement to locate the desired RID, and resubmit the request in a new utility job step.

DSNU659I *csect-name* **REPAIR LEVELID OPERATION DISALLOWED.**

Explanation: The REPAIR utility with the LEVELID clause is not allowed. Possible reasons are:

- The data set was in use at the time the REPAIR utility was run.
- The page set has outstanding log apply activity (an indoubt UR has interest in the page set or the page set has pages in the logical page list (LPL)).

Severity: 8 (error)

System action: Processing ends with this operation.

User response: The REPAIR LEVELID operation for the given index space or table space cannot continue at this stage. If the page set is in use, you should not attempt to run REPAIR LEVELID. If the page set has log apply activity outstanding (in-doubt UR or pages in LPL), the log apply activity must complete before using REPAIR LEVELID. Submit other REPAIR control statements in a different job step.

DSNU660I *csect-name* **POSSIBLE RID** *rid*

Explanation: 'rid' is one of several records that map to the specified KEY.

Severity: 8 (error)

System action: Other records also mapping to the specified KEY are identified by other DSNU660I messages. This message is preceded by message DSNU658I. Processing continues normally.

User response: Change the LOCATE statement to locate the desired RID, and resubmit the request in a new utility job step. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNU661I *csect-name* **RECORD NOT FOUND IN PAGE, MAXID=** *hex-data*

Explanation: A RID was either specified or obtained from the clustering index via the specified KEY. The PAGE indicated in the RID did not contain the RECORD indicated in the RID.

Severity: 8 (error)

System action: Processing proceeds to the next LOCATE statement, if there is one.

User response: Either change the LOCATE statement to locate a valid RID or KEY, or use LOCATE PAGE and resubmit the REPAIR statements in a new utility job step. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNU662I *csect-name* **NO RECORD FOUND WITH SPECIFIED KEY**

Explanation: A KEY was specified on the LOCATE statement, but no entry exists in the clustering index for that KEY.

Severity: 8 (error)

System action: Processing begins at the next LOCATE statement, if there is one.

User response: Correct the LOCATE statement, and resubmit the REPAIR statements in a new utility job step.

DSNU663I *csect-name* **PAGE** *hex-page-number* **NOT FOUND**

Explanation: The page identified a record identifier (RID) or KEY does not exist in the specified table space.

Severity: 0 (informational) when issued by DUMP PAGES(n). 8 (error) when issued by LOCATE PAGE(n).

System action: Processing begins at the next LOCATE statement, if there is one.

User response: Correct the LOCATE statement, and resubmit the REPAIR statements in a new utility job step.

DSNU664I *csect-name* **INVALID OFFSET OR LENGTH FOR** *xxxxx*

Explanation: 'xxxxx' is 'RECORD' or 'PAGE'. The specified or default values for OFFSET and LENGTH combine to extend beyond the specified record or page.

Severity: 4 (warning)

System action: Processing continues at the next LOCATE statement, if there is one.

User response: Correct the OFFSET and/or LENGTH specification, and resubmit the REPAIR statements in a new utility job step. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNU665I *csect-name* **NO RECORD ASSOCIATED WITH SPECIFIED RID**

Explanation: A record identifier (RID) was specified on the LOCATE statement, but that RID is not associated with any record in the specified table space. If DELETE processing returns this message, the RID may exist but be invalid for DELETE.

Severity: 8 (error)

System action: Processing begins at the next LOCATE statement, if there is one. Subsequent utilities in the same job step will not be executed.

User response: Correct the LOCATE statement, and resubmit the REPAIR statements in a new utility job step.

Problem determination: REPAIR DELETE will issue this message for RIDs which are internal and do not directly represent user data. The DUMP keyword may be used to view the RID, if it exists. These internal records may only be repaired using the REPLACE keyword.

DSNU666I *csect-name* **NO INDEX EXISTS**

Explanation: The KEY for this table space is not a full clustering key. The statement is invalid.

Severity: 8 (error)

System action: Processing continues at the next LOCATE statement, if there is one.

User response: Correct the LOCATE statement to use the clustering index, or change LOCATE by PAGE or RID. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNU668I *csect-name* **UNABLE TO CONVERT KEY TO** *format*

Explanation: 'format' specifies the data type to which the KEY specification must be converted. The conversion has failed.

Severity: 8 (error)

System action: Processing begins at the next LOCATE statement, if there is one.

User response: Correct the LOCATE statement to specify a KEY either of the indicated data type or of a compatible data-type. Then resubmit the REPAIR statements in a new utility job step. Refer to *DB2 Utility Guide and Reference* for a discussion of conversion constraints.

DSNU669I *csect-name* **INDEX NOT FOUND FOR SPECIFIED TABLESPACE**

Explanation: An index was specified on the LOCATE KEY statement via the INDEX keyword, but no index by that name was found related to the specified TABLESPACE.

Severity: 8 (error)

System action: Processing begins at the next LOCATE statement, if there is one.

User response: Correct the LOCATE statement and resubmit the REPAIR statements in a new utility job step.

Problem determination: Valid index names can be found in the SYSIBM.SYSINDEXES catalog table.

DSNU670I *csect-name* **PAGE X** *page-id* **IS A BROKEN PAGE**

Explanation: The indicated page located by the REPAIR utility has been marked broken by the data manager. The 'page id' is given in hexadecimal. This page is not accessible until the broken page indicator is reset by the REPAIR REPLACE RESET option.

Severity: 4 (warning)

System action: The message reports that the page in which the REPAIR utility is requested to REPLACE some data is broken.

User response: Determine the cause of the broken page with the assistance of the system programmer. When it is repaired, use the REPAIR RESET option.

System programmer response: Determine the cause of the broken page. It could be an internal error and require support center assistance. The REPAIR DUMP option can be used to determine the contents of the page for failure analysis. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNU672I *csect-name* **INDEX USED HAS MULTIPLE-FIELD KEY**

Explanation: A REPAIR LOCATE KEY operation was attempted against a table space using an index that has a multiple-field key. LOCATE KEY does not support multiple-field keys.

Severity: 8 (error)

System action: Processing begins at the next LOCATE scope, if any.

User response: Change the LOCATE statement to specify PAGE and RID, and resubmit the REPAIR statements in a new utility job step.

DSNU673I *csect-name* **REPAIR LOCATE RID LOCATED A NON-DATA PAGE**

Explanation: The LOCATE RID option of the REPAIR utility was used to process a nondata page.

Severity: 8 (error)

System action: Processing continues at the next LOCATE statement. Otherwise, processing stops.

User response: To process nondata pages, use the LOCATE PAGE option in the REPAIR control statements and resubmit the new utility job step. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNU676I *csect-name* **- ROWID=***rowid* **VERSION=***version* **IS NOT FOUND IN AUXILIARY INDEX**

Explanation: The REPAIR utility was unable to locate the ROWID *rowid* and VERSION *version* in the index on the auxiliary table.

System action: Utility returns completion code=8.

Programmer response: Correct the ROWID or VERSION. Only ROWID and VERSION values previously generated by DB2 can be used. The most likely source values of a ROWID and VERSION are messages DSNU743I, DSNU805I, and DSNU806I from the CHECK DATA or CHECK LOB utilities.

DSNU678I *csect-name* **VERIFY, REPLACE, DUMP, OR DELETE BEFORE LOCATE**

Explanation: The REPAIR utility was requested to VERIFY, REPLACE, or DUMP or DELETE a page or record and that page of record could not be located.

Severity: 8 (error)

System action: Processing begins at the next LOCATE statement, if there is one. Subsequent utilities in the same job step will not be executed.

User response: Correct the REPAIR control statements, and resubmit the request in a new utility job step. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNU679I *csect-name* **MULTIPLE 'DELETE'S PER 'LOCATE' NOT ALLOWED**

Explanation: The REPAIR utility was requested to DELETE the same record more than once. Only one DELETE is allowed per LOCATE RID or LOCATE KEY.

Severity: 8 (error)

System action: Processing begins at the next LOCATE scope, if any. Subsequent utilities in the same job step are not executed.

User response: Remove all but one DELETE control statement from each LOCATE scope which is intended to delete a record. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNU680I *csect-name* **PAGE** *hex-page-number*

Explanation: This message is issued as an identifier at the top of each page dumped by the REPAIR DUMP PAGES facility. 'hex-page-number' identifies the page which is about to be dumped.

Severity: 0 (informational)

System action: Processing continues.

DSNU681I *csect-name* **SET NOCHECKPEND OPERATION SUCCESSFUL**

Explanation: This message reports the successful completion of the REPAIR SET TABLESPACE NOCHECKPEND utility operation. The check pending restriction has been removed from the specified table space.

Severity: 0(informational)

System action: The REPAIR SET statement has completed normally.

DSNU682I *csect-name* **SET NORCVRPEND OPERATION SUCCESSFUL**

Explanation: This message reports the successful completion of the REPAIR SET NORCVRPEND utility operation. The RECOVERY PENDING restriction has been removed from the specified table space, from the specified index, or from all indexes of the specified table space.

Severity: 0 (informational)

System action: The REPAIR SET statement has completed normally.

DSNU683I *csect-name* **REPAIR LEVELID OPERATION SUCCESSFUL.**

Explanation: The REPAIR utility with the LEVELID clause completed successfully.

Severity: 0 (informational)

System action: Processing completed successfully.

User response: Submit any other REPAIR control statements in a different job step.

DSNU684I *csect-name* **- SET NOAUXCHKP OPERATION SUCCESSFUL**

Explanation: This message reports the successful completion of the REPAIR SET NOAUXCHKP utility operation. The auxiliary check pending (ACHKP) state has been removed for the specified base table space.

System action: Utility returns completion code=0.

DSNU685I *csect-name* **SET NORBDPEND OPERATION SUCCESSFUL**

Explanation: This message reports the successful completion of the REPAIR SET NORBDPEND utility operation. The three restrictions REBUILD PENDING, REBUILD PENDING STAR and PAGESET REBUILD PENDING have been removed from the specified index or from all indexes of the specified table space.

Severity: 0 (informational)

System action: The REPAIR SET statement has completed normally.

DSNU686I *csect-name* **- SET NOAUXWARN OPERATION SUCCESSFUL**

Explanation: This message reports the successful completion of the REPAIR SET NOAUXWARN utility operation. The AUX WARNING state has been removed for the specified table space.

Severity: 0 (informational)

System action: Utility returns completion code=0.

Programmer response: None.

DSNU687I *csect-name* **- ROWID=***rowid* **VERSION=***version* **LOB DELETED**

Explanation: The LOB with the ROWID *rowid* and VERSION *version* was successully deleted from the specified LOB table space.

System action: Utility returns completion code=0.

Programmer response: None

DSNU688I *csect-name* **SET NOARBDPEND OPERATION SUCCESSFUL**

Explanation: The REPAIR utility successfully reset an object that was in an advisory rebulid pending (ARBD) state.

Severity: 0 (informational)

System action: No action required.

User response: No response required.

DSNU689I *csect-name* **SET NOAREORPENDSTART OPERATION SUCCESSFUL**

Explanation: The REPAIR utility successfully reset an object that was in an advisory reorg pending (AREO*) + state.

Severity: 0 (informational)

System action: No action required.

User response: No response required.

DSNU691I *csect-name* **- LOB WITH ROWID=***rowid* **VERSION=***version* **NOT FOUND**

Explanation: The REPAIR utility was unable to find a LOB with the ROWID of *rowid* and VERSION *version*.

System action: Utility returns completion code=8.

Programmer response: Either the ROWID is incorrect, or the index on the auxiliary table is broken. In the first case correct the ROWID as specified on the **LOCATE ROWID** statement. If the index on the auxiliary table is

broken, execute the REBUILD INDEX utility to rebuild the index.

DSNU692I *csect-name* **A REQUIRED REORG HAS NOT BEEN DONE**

Explanation: This message is issued for the following reasons:

- The REPAIR VERSION utility detected that a REORG was not run on the object after the last ALTER that created a new version.
- The REPAIR VERSION utility detected that a REORG was not run on the object after the last ALTER that created a new version.

Severity: 4 (warning)

System action: The REPAIR VERSION utility continues to run. Objects are still usable, but consider performing REORG on the object.

User response: Refer to the, "Updating version information when moving objects to another subsystem" in the REPAIR topic in *DB2 Utility Guide and Reference* for more information about the steps that you need to follow when you move objects to another subsystem.

DSNU700I *csect-name nnnn* **INDEX ENTRIES UNLOADED FROM INDEX=***iiii* **PARTITION=***pppp*

Explanation: The CHECK utility UNLOAD phase unloaded nnnn records from the index named iiii and partition pppp.

Severity: 0 (informational)

System action: Processing continues.

DSNU701I *csect-name nnnn* **INDEX ENTRIES UNLOADED FROM** *iiii*

Explanation: The CHECK Utility UNLOAD phase unloaded *nnnn* records from the index named *iiii*.

Severity: 0 (informational)

System action: Processing continues.

DSNU702I *csect-name* **INVALID INDEX SPECIFICATION. INDEX=***iiii*

Explanation: All tables referenced by the INDEXES in the CHECK or REBUILD utility must be in the same table space. Index 'iiii' is not over a table in the table space specified with the TABLESPACE keyword.

Severity: 8 (error)

System action: Utility processing is discontinued.

User response: Correct the index name and/or the TABLESPACE parameters such that a single table space is involved, and rerun the utility job.

Problem determination: To CHECK or REBUILD ALL INDEXES in a table space, use the TABLESPACE keyword, but do not use the list option. This is one way to correct this error. If you want to correct the list of indexes specified in the index name list, the following procedure will locate the table space name related to the index 'iiii'.

The index name 'iiii' reported in the message is of the form 'creator.name'. Obtain the TBCREATOR and TBNAME columns from the DB2 catalog table SYSIBM.SYSINDEXES where columns CREATOR and NAME have the values reported in the message. Obtain the DBNAME and TSNAME columns from the DB2 catalog table SYSIBM.SYSTABLES, where columns CREATOR and NAME have the values obtained from TBCREATOR and TBNAME of SYSIBM.SYSINDEXES.

The values for DBNAME and TSNAME must be the same for all indexes specified in the index name list and must be the same as the table space name specified on the TABLESPACE keyword (if provided).

DSNU703I *csect-name* **INVALID INDEX SPECIFICATION. INDEX=***iiii*

Explanation: All tables referenced by the indexes in the CHECK or REBUILD utility must be in the same table space. Index 'iiii' is not over a table in the same table space as the prior indexes listed in the index name list.

Severity: 8 (error)

System action: Utility processing is discontinued.

User response: Correct the index name and/or the TABLESPACE parameters such that a single table space is involved, and rerun the utility job.

Problem determination: To CHECK or REBUILD ALL INDEXES in a table space, use the TABLESPACE keyword, but do not use the list option. This is one way to correct this error. If you want to correct the list of indexes specified in the index name list, the following procedure will locate the table space name related to the index 'iiii'.

The index name 'iiii' reported in the message is of the form 'creator.name'. Obtain the TBCREATOR and TBNAME columns from the DB2 catalog table SYSIBM.SYSINDEXES, where columns CREATOR and NAME have the values reported in the message. Obtain the DBNAME and TSNAME columns from the DB2 catalog table SYSIBM.SYSTABLES, where columns CREATOR and NAME have the values obtained from TBCREATOR and TBNAME of SYSIBM.SYSINDEXES.

The values for DBNAME and TSNAME must be the same for all indexes specified in the index name list and must be the same as the table space name specified on the TABLESPACE keyword if provided.

DSNU705I *csect-name* **UNLOAD PHASE COMPLETE, ELAPSED TIME=***tttt*

Explanation: The unload phase of the CHECK or REBUILD INDEX utility is complete.

Severity: 0 (informational)

System action: Processing continues.

DSNU706I *csect-name* **ROW IS IN WRONG PARTITION RID=***rrrr*, **INDEX=***iiii*, **KEY=***kkkk*, **CORRECT PART=***pppp*, **WRONG PART=***wwww*

Explanation: The row at RID 'rrrr' resides in the wrong partition. The partitioning KEY 'kkkk' for partitioning INDEX 'iiii' indicates that the row at RID 'rrrr' should reside in partition 'pppp', however the rid indicates that the row contains partition 'wwww'.

Severity: 4 (warning)

System action: Utility processing continues.

User response: Run RECOVER TABLESPACE utility to recover the data or run REPAIR DELETE utility to delete the row 'rrrr' followed by CHECK INDEX and CHECK DATA utility if referential constraints are involved.

Problem determination: The inconsistency between the table space partition number of a row and the index space partition number of the row's clustering key may be due to a DB2 internal error, but is most likely due to a user using the DSN1COPY service aid to populate a target table space with data from a source table space when the clustering key values on the source and target table spaces are not identical.

DSNU707I *csect-name* **INDEX ENTRY POINTS TO WRONG TABLE. OBID=***oooo*, **RID=***rrrr*, **INDEX=**

Explanation: The CHECK utility CHECKIDX phase found that index 'iiii' contains an entry for key 'kkkk' that points to record at RID 'rrrr', but the record belongs to the table with OBID 'oooo'.

Severity: 4 (warning)

System action: CHECK utility processing continues.

User response: Run the REBUILD INDEX utility for index 'iiii'.

DSNU708I *csect-name* **INDEX IS NOT UNIQUE. INDEX=***iiii*, **KEY=***kkkk*

Explanation: The CHECK utility CHECKIDX phase found that unique index 'iiii' contains a nonunique key, 'kkkk'.

Severity: 4 (warning)

System action: CHECK utility processing continues.

User response: Drop the unique index, and delete the duplicate rows. Re-create the index.

DSNU709I *csect-name* **NO INDEX ENTRY FOR ROW. RID=***rrrr*, **INDEX=***iiii*, **KEY=***kkkk*

Explanation: The CHECK utility CHECKIDX phase found that index *iiii* is missing an entry for the row at RID *rrrr* with key *kkkk*.

Severity: 4 (warning)

System action: CHECK utility processing continues.
The utility will complete with return code 4.

User response: Run the REBUILD INDEX utility for index *iiii*.

DSNU710I *csect-name* **INDEX ENTRY CONTAINS INVALID ZERO REPRESENTATION. RID=***rrrr*, **INDEX=***iiii*, **KEY=***kkkk*

Explanation: The CHECK utility CHECKIDX phase found that index 'iiii' has an entry that contains a negative zero. The index entry is for the row at RID 'rrrr' with key 'kkkk'.

Severity: 4 (warning)

System action: CHECK utility processing continues.

User response: Run the REBUILD INDEX utility for index 'iiii'.

DSNU711I *csect-name* **NO ROW FOR INDEX ENTRY. RID=***rrrr*, **INDEX=***iiii*, **KEY=***kkkk*

Explanation: The CHECK utility CHECKIDX phase found that the index 'iiii' contains an entry that has no row. The index entry contained RID 'rrrr' and key 'kkkk', but RID 'rrrr' is not a valid record.

Severity: 4 (warning)

System action: CHECK utility processing continues.

User response: Run the REBUILD INDEX utility for index 'iiii'.

DSNU712I *csect-name* **INDEX OUT OF SEQUENCE. INDEX=***iiii*, **KEY=***kkkk*

Explanation: The CHECK utility CHECKIDX phase found that index 'iiii' is not in correct sequence.

Severity: 4 (warning)

System action: CHECK utility processing continues.

User response: Run the REBUILD INDEX utility for index 'iiii'.

DSNU713I *csect-name* **KEYS MISMATCH. RID=***rrrr***, INDEX=***iiii***, KEY=***kkkk***, DATA KEY=***jjjj***

Explanation: The CHECK utility CHECKIDX phase found that index iiii contains an entry for the row at rid rrrr that is inconsistent. KEY kkkk does not match DATA KEY jjjj.

Severity: 4 (warning)

System action: CHECK utility processing continues.

User response: Run the REBUILD INDEX utility for index 'iiii'.

DSNU715I *csect-name* **MULTIPLE INDEX ENTRIES FOR ROW. RID=***rrrr***, INDEX=***iiii***, KEY=***kkkk***

Explanation: The CHECK utility CHECKIDX phase found that index 'iiii' has multiple index entries for the row at RID 'rrrr' with key 'kkkk'.

Severity: 4 (warning)

System action: CHECK utility processing continues.

User response: Run the REBUILD INDEX utility for index 'iiii'.

DSNU717I *csect-name nnnn* **ENTRIES CHECKED FOR INDEX=***iiii* **PARTITION=***pppp*

Explanation: The CHECK utility completed checking index iiii partition pppp and processed nnnn index entries. This message is issued once for each index partition checked.

Severity: 0 (informational)

System action: Processing continues.

DSNU718I *csect-name* **NO INDEXES FOUND FOR TABLESPACE** *tttt*

Explanation: A utility has been invoked with the TABLESPACE keyword specifying a table space containing tables that have no indexes.

Severity: 4 (warning)

System action: This message indicates completion of the utility. Processing continues to the next utility statement in the job step, if any. If there are no utility statements following, the job step is terminated normally and the UTILID of the job step becomes available for reuse.

User response: If the table space specification is in error, correct it and resubmit the utility. If the table space specification is correct, and the table space contains no indexed tables, this is a normal indication and no response is necessary.

DSNU719I *csect-name nnnn* **ENTRIES CHECKED FOR INDEX** *iiii*

Explanation: The CHECK utility has completed checking index 'iiii' and has processed 'nnnn' index entries. This message will be issued once for each index checked.

Severity: 0 (informational)

System action: Processing continues.

DSNU720I *csect-name* **CHECKIDX PHASE COMPLETE, ELAPSED TIME =** *tttt*

Explanation: The CHECK utility CHECKIDX phase is complete.

Severity: 0 (informational)

System action: Processing continues.

DSNU721I *csect-name* **- USER** *auth-id* **DOES NOT HAVE DELETE AUTHORITY ON TABLE**

Explanation: The CHECK DATA utility was run with DELETE YES, but user *auth-id* does not have authority to delete from a table specified in the utility statement or from one of its descendents.

Severity: 8 (error)

System action: Further utility processing is terminated.

User response: Obtain authorization to delete from the tables specified in the utility statement and any of its descendents that are affected.

DSNU722I *csect-name* **- USER** *userid* **DOES NOT HAVE INSERT AUTHORITY ON EXCEPTION TABLE**

Explanation: The CHECK DATA utility was run with FOR EXCEPTION specified but user 'userid' does not have authority to insert into an exception table specified in the utility statement.

Severity: 8 (error)

System action: Further utility processing is terminated.

User response: Obtain authorization to insert into all of the exception tables specified in the utility statement.

DSNU725I *csect-name* **DUPLICATE TABLESPACE** *tsp* **NAMES EXIST IN THE TABLESPACE LIST**

Explanation: This message is issued by the CHECK DATA utility when a duplicate table space 'tsp' is found in the table space list. An entry in the list for the same table space with a partition specified is also considered

a duplicate. This is because all partitions are the default when no partition is specified for a partitioned table space. The CHECK DATA utility proceeds and the duplicate entry is skipped.

Severity: 4 (warning)

System action: Processing continues.

DSNU726I *csect-name* **SCOPE PENDING IS ACTIVE, BUT THE ENTIRE TABLE SPACE** *tsp* **MUST BE CHECKED**

Explanation: This message is issued by the CHECK DATA utility when checking with the option SCOPE PENDING. Table space *tsp* is in check pending status, but the utility is checking all rows in dependent tables or tables defined with check constraints in that table space.

Severity: 4 (warning)

System action: The utility continues processing.

DSNU727I *csect-name* **TABLESPACE** *tsp* **IS NOT CHECK PENDING**

Explanation: This message is issued when 'CHECK DATA TABLESPACE tsp SCOPE PENDING' is specified and the CHECK DATA utility finds that the table space is not in a check pending state. The CHECK DATA utility skips table space 'tsp' and will continue with any other table space specified. The catalog fields that reflect the extent of a check pending state are updated to ensure that they indicate that no check is required.

Severity: 4 (warning)

System action: Processing continues.

DSNU728I *csect-name* **INVALID EXCEPTION TABLE** *table2* **FOR** *table1*

Explanation: This message is issued when the CHECK DATA utility finds an exception table is invalid. The exception table named 'table2' which was specified for rows from table, 'table1', is invalid. Either 'table2' was missing columns of 'table1' or 'table2' contained incompatible column attributes. The CHECK DATA utility will terminate processing after all exception tables have been examined.

Severity: 8 (error)

System action: The CHECK DATA utility will terminate processing after all the exception tables have been examined.

User response: Refer to Part 2 (Volume 1) of *DB2 Administration Guide* for a definition of the CREATE TABLE command with the LIKE operand to correct the invalid exception table.

DSNU729I *csect-name* **TABLE** *table-name* **NOT SPECIFIED IN THE 'FOR EXCEPTION' CLAUSE**

Explanation: The CHECK DATA utility did not find a required table listed in the FOR EXCEPTION clause. All tables with constraints in the table spaces being checked and their descendents must be listed in the FOR EXCEPTION clause of CHECK DATA utility.

Severity: 8 (error)

System action: The CHECK DATA utility terminates processing after all the required tables are examined.

User response: Examine the utility input statement to determine the reason for the missing table named in the message. Correct the input statement to include the missing table and resubmit the job. Refer to *DB2 Utility Guide and Reference* for information about the FOR EXCEPTION clause of the CHECK DATA utility and use of exception tables.

DSNU730I *csect-name* **CHECKING TABLE** *tab*

Explanation: This message is issued when the CHECK DATA utility begins to process a table. The name of table being checked is 'tab'.

Severity: 0 (informational)

System action: Processing continues.

DSNU731I *csect-name* **EXCEPTION LIMIT REACHED**

Explanation: The EXCEPTION limit specified on the CHECK utility statement has been reached.

Severity: 8 (error)

System action: The CHECK utility terminates processing.

User response: If required, increase the exception limit and resubmit the job.

DSNU732I **NO TABLE TO BE CHECKED FOR TABLESPACE** *tsp*

Explanation: The CHECK DATA utility determined that none of the tables in table space *tsp* has a constraint defined. Therefore, no table in that table space is checked.

This message is issued by the following CSECT: DSNUKINP

Severity: 4 (warning)

System action: Utility processing continues.

DSNU733I *csect-name* **ROW (RID=***rrrrrr***) HAS NO PARENT FOR** *rel*

Explanation: This message is issued when the CHECK DATA utility finds a row in a dependent table without a parent. The row with RID, 'rrrrrr', has no parent for the relationship named 'rel'. This message is always proceeded by a DSNU730I message that identifies the table in error.

Severity: 4 (warning)

System action: Processing continues.

User response: Refer to Part 2 (Volume 1) of *DB2 Administration Guide* for a discussion of the check pending operation.

DSNU735I *csect-name* **DUPLICATE TABLESPACE** *tsp* **PARTITION** *part* **NAMES EXIST IN THE TABLESPACE LIST**

Explanation: This message is issued by the CHECK DATA utility when a duplicate table space 'tsp' and partition 'part' is found in the table space list. An entry in the table space list with a partition specified is considered to be a duplicate when another entry for the same table space with no partition specified (defaults to all partitions for a partitioned table space) is found. In this case, the table space list entry with no partition specified has priority and the entry with a partition specified is skipped. The CHECK DATA utility proceeds and the duplicate entry is skipped.

Severity: 4 (warning)

System action: Processing continues.

DSNU736I *csect-name* **SCOPE PENDING IS ACTIVE, BUT THE ENTIRE TABLE SPACE** *tsp* **PARTITION** *part* **MUST BE CHECKED**

Explanation: This message is issued by the CHECK DATA utility when checking with the option SCOPE PENDING. Table space *tsp* partition *part* is in check pending status, but the utility is checking all rows in that table space partition.

Severity: 4 (warning)

System action: The utility continues processing.

DSNU737I *csect-name* **TABLESPACE** *tsp* **PARTITION** *part* **IS NOT CHECK PENDING**

Explanation: This message is issued when 'CHECK DATA TABLESPACE tsp PART part SCOPE PENDING' is specified and the CHECK DATA utility finds that the table space partition is not in a check pending state. The CHECK DATA utility skips table space 'tsp' partition 'part' and continues with any other table space specified. The catalog fields that reflect the extent of a check pending state are updated to ensure that they indicate that no check is required.

Severity: 4 (warning)

System action: Processing continues.

DSNU739I *csect-name* **CHECK TABLE** *tab* **COMPLETE,** *elapsed time=t1*

Explanation: This message is issued when the CHECK DATA utility finishes processing a table. The elapsed clock time that the check of table 'tab' consumed is 't1'.

Severity: 0 (informational)

DSNU740I *csect-name* **CHECKING TABLE** *tab* **PARTITION** *part*

Explanation: This message is issued when the CHECK DATA utility begins to process a table of a partitioned table space in which a single partition was specified. The name of the table and the number of the partition being checked are identified in the message.

Severity: 0 (informational)

System action: Processing continues.

DSNU741I *csect-name nn* **ROWS DELETED FROM TABLE** *tab*

Explanation: This message is issued by the CHECK DATA utility when the DELETE YES option is specified. The number of rows deleted from TABLE *tab* is the number of rows that contain referential integrity violations or check constraint violations, plus their descendent rows. The number of rows deleted from TABLE *tab* is equal to the number of rows inserted into the corresponding exception table.

Severity: 4 (warning)

System action: Processing continues.

DSNU742I *csect-name* **ROW (RID=***rrrrrr***) VIOLATES CHECK CONSTRAINT** *constraint-name* **ON TABLE** *table-name*

Explanation: The CHECK DATA utility found that the row with RID=*rrrrrr* in table *table-name* violated check constraint *constraint-name*.

Severity: 4 (warning)

System action: Processing continues

User response: Refer to Part 2 (Volume 1) of *DB2 Administration Guide* for a discussion of the check pending operation and recovery procedures.

DSNU743I *csect-name* **- LOB IS INVALID. ROWID=***rowid* **VERSION=***version* **IS INVALID**

Explanation: The CHECK LOB utility found that the LOB with ROWID *rowid* and VERSION *version* was marked as invalid.

Severity: 4

System action: The LOB table space status is set to auxiliary warning (AUXW).

User response: You must use SQL to update the LOB column or delete the row that contains the LOB column to remove the invalid indication. REPAIR LOCATE ROWID VERSION DELETE can be used to remove the entire LOB from the LOB table space.

DSNU745I *csect-name* **- ROWID=***rowid* **VERSION=***version* **RID=***rid* **MISSING IN LOB TABLE SPACE=***lob-table-space-name*

Explanation: The CHECK INDEX utility found an index entry key with ROWID *rowid* and VERSION *version* in the index, but not in the LOB table space named *lob-table-space-name*.

System action: The utility returns completion code=4. The index is set Recover Pending.

Programmer response: Use REBUILD INDEX or RECOVER to correct the index on the auxiliary table.

DSNU747I *csect-name* **- ROWID=***rowid* **VERSION=***version* **MISSING IN INDEX=***index-name*

Explanation: The CHECK INDEX utility found a LOB with ROWID *rowid* and VERSION *version* in the LOB table space but not in the index on the auxiliary table, *index-name*.

System action: Utility returns completion code=4. The index is set Recover Pending.

Programmer response: Use REBUILD INDEX to correct the index on the auxiliary table.

DSNU749I *csect-name* **CHECK DATA COMPLETE, ELAPSED TIME=***t1*

Explanation: This message is issued when the CHECK DATA utility is complete. The elapsed clock time that the CHECK DATA utility consumed is 't1'.

Severity: 0 (informational)

System action: Processing ends.

DSNU750I *csect-name* **CATMAINT** *operation* **PHASE** *phase-number* **STARTED**

Explanation: The CATMAINT utility phase *phase-number* processing started for the indicated operation.

Severity: 0 (informational)

System action: Utility processing continues normally.

DSNU751I *csect-name* **CATALOG** *operation* **OF TABLE -** *table-name*

Explanation: This message is issued the first time data is operated upon for the designated table.

Severity: 0 (informational)

System action: Utility processing continues normally.

DSNU752I *csect-name* **CATMAINT** *operation* **PHASE** *phase-number* **COMPLETED**

Explanation: The CATMAINT utility PHASE *phase-number* processing completed for the indicated operation.

Severity: 0 (informational)

System action: Utility processing continues normally.

DSNU756I *csect-name* **CATALOG TABLE IS EMPTY -** *table-name*

Explanation: This message is issued when the named table is empty.

Severity: 0 (informational)

System action: Utility processing continues normally.

DSNU758I *csect-name* **- WARNING DURING PROCESSING OF SQL STATEMENT - SQLCODE IS:** *sql-code* **SQL MESSAGE TEXT:** *message-text***. SQL STATEMENT:** *sql-statement*

Explanation: CATMAINT executed an SQL statement as part of CATMAINT UPDATE processing. DB2 returned a non-zero SQLCODE. This message does not indicate a problem in the CATMAINT process.

Severity: 0 (informational)

System action: Utility processing continues normally.

DSNU759I *csect-name* **INCORRECT UNLOAD DATASET HAS BEEN DETECTED**

Explanation: This message indicates an inconsistency between the release level of the unloaded catalog and the DB2 release that is running the CATMAINT utility.

Severity: 8 (error)

System action: Utility processing terminates.

User response: Review the JCL for the job to ensure that the correct unload data set is specified. Verify that the correct level of DB2 is up.

DSNU760I *csect-name* **SPECIFIED USER NOT DEFINED WITH INSTALLATION SYSADM AUTHORITY**

Explanation: The utility can be run only under a user defined with the installation SYSADM authority.

Severity: 8 (error)

System action: Utility processing terminates.

User response: Rerun the utility with the proper authorization.

DSNU762I *csect-name* **DUPLICATE RECORD DETECTED FOR TABLENAME -** *table-name* **SEQNO OF THE INPUT RECORD IS -** *seqno* **FOLLOWING IS UP TO 119 BYTES OFRECORD BEING INSERTED**

Explanation: An object created in the target catalog has the same name as an object being migrated from the source catalog.

Severity: 8 (error)

System action: Utility abends.

User response: User data should not be in the target catalog before running the CATMAINT utility. Free or drop the object from the target catalog and rerun the CATMAINT utility.

DSNU764I **csect-name FOLLOWING IS UP TO 119 BYTES OF RECORDFOR TABLE table-name, SEQNO seqno BEING PROCESSED AT THE TIME OF ERROR**

Explanation: An abend occurred during the execution of the CATMAINT utility, during the processing of the table, 'table-name', with a sequence number of 'seqno'. The cause of the abend is one of the following:

- The record sequence number was invalid.
- An unexpected return code was received from DSNUEXLO.
- The table name was not found in the load table (DSNUELTB).
- The parent record could not be positioned.
- The position OBID was zero (DSNUELTB has incorrect value).
- The record action code is incorrect.

Following this message, up to the first 119 bytes of the recordin question is displayed.

Severity: 8 (error)

System action: Utility abends.

User response: Verify that the record in the unloaded data set is correct.

DSNU765I *csect-name* **THE DB2 CATALOG IS NOT AT THE REQUIRED LEVEL. CATALOG NOT UPDATED**

Explanation: The DB2 catalog is not at the proper catalog level to migrate the catalog. This error can occur due to the following reasons:

- Attempting to migrate the catalog from a DB2 release that does not support migration
- Attempting to migrate DB2 Stored Procedures prior to a catalog migration

Severity: 8 (error)

System action: Utility processing terminates

User response: Depending on the migration error, perform the necessary actions:

- If the catalog is not at the correct level for a migration, fall back to the prevoius release. Migrate only from DB2 releases that support migration.
- Prior to attempting a Stored Procedure migration, migrate the DB2 catalog to the current release. After migrating the DB2 catalog, then you can migrate the Stored Procedures.

DSNU766I *csect-name* **THE CATALOG HAS ALREADY BEEN MIGRATED. NO ACTION WILL BE TAKEN**

Explanation: The catalog was already successfully migrated. There is no reason to rerun the catalog migration job. Therefore, no action is taken to update the DB2 catalog.

Severity: 0 (informational)

System action: Processing ends.

DSNU767I *csect-name* **DB2 SYSTEM MUST BE STARTED ACCESS(MAINT)**

Explanation: In order for the utility to execute, the DB2 subsystem must be started with ACCESS(MAINT). This message indicates that the subsystem was not started in this manner.

Severity: 8 (error)

System action: -STOP DB2, and then -START DB2 with ACCESS(MAINT).

DSNU768I *csect-name* **CATMAINT CONVERT COMPLETE FOR INDEX** *index-name*

Explanation: The conversion is complete for index *index-name*.

Severity: 0 (informational)

System action: Processing continues with the next index.

DSNU769I *csect-name* **THE BUILD PHASE FAILED FOR INDEX** *index-name* **REASON** *reason*

Explanation: The CATMAINT CONVERT build phase failed for index *index-name*. This message and others identify the actions that should be taken.

Severity: 8 (error)

Operator response: Note the reason code and the index name. See the Problem Determination section for an explanation of the collected data.

System action: Utility processing continues with the next index and will terminate with return code 8.

User response: Correct the problem indicated by the reason code. Either resubmit the CATMAINT CONVERT job or recover the index that failed.

Problem determination: The reason the build phase failed is identified by the reason code, which is described in *DB2 Codes*.

DSNU770I *csect-name* **- DBD** *type* **STARTED FOR DATABASE** *database*

Explanation: This message marks the start of the REPAIR DBD type utility. In the message, 'type' can be DIAGNOSE, REBUILD, or TEST. The database name specified in the REPAIR statement is 'database'.

Severity: 0 (informational)

System action: Utility processing continues normally.

DSNU771I *csect-name* **- DBD** *type* **COMPLETE FOR DATABASE** *database*

Explanation: This message marks the completion of the REPAIR DBD type utility. In the message, 'type' can be DIAGNOSE, REBUILD, or TEST. The database name specified in the REPAIR statement is 'database'.

Severity: 0 (informational)

System action: Utility processing continues normally.

DSNU772I *csect-name* **- THE** *object-type qualifier.object-name* **IS UNAVAILABLE IN THIS RELEASE**

Explanation: This message indicates that the specified object has a dependency on a release from which fall back has occurred, and cannot be operated upon in the current release.

Severity: 8 (error)

System action: Utility processing terminates.

User response: Check the IBMREQD column of the catalog table associated with the specified object type. If the value in the catalog table is the release dependency mark for a release of DB2 from which fall back has occurred, the utility is not operational for the current release of the targeted database.

DSNU773I *csect-name* **-** *database* **IS A SYSTEM DATABASE AND CANNOT BE PROCESSED**

Explanation: The database indicated is not eligible for processing by the type of the REPAIR DBD utility involved.

Severity: 8 (error)

System action: Utility processing terminates.

User response: Ensure that the utility control statement specifies the correct database.

DSNU774I *csect-name* **- SPECIFIED DATABASE** *database* **NOT FOUND**

Explanation: The database indicated by the utility control statement was not found in catalog table SYSIBM.SYSDATABASE.

Severity: 8 (error)

System action: Utility processing terminates.

User response: Ensure that the database indicated by the utility control statement exists in SYSIBM.SYSDATABASE.

DSNU775I *csect-name* **- DROP OF DATABASE** *dbname* **FAILED WITH SQLCODE =** *sqlcode*

Explanation: An error was encountered while using the REPAIR DBD DROP utility to drop DATABASE 'dbname'. The SQL error code is indicated by 'sqlcode'.

Severity: 8 (error)

System action: DATABASE 'dbname' is not dropped.

User response: For a description of the SQL error code that was issued, refer to *DB2 Codes*.

DSNU776I *csect-name* **CATMAINT** *operation* **ERROR -** *error-text*

Explanation: The CATMAINT utility halted because the catalog contains view(s) on the SYSCOLDIST and/or SYSCOLDISTSTATS catalog tables.

Severity: 0 (informational)

System action: Utility processing halts.

User response: Remove the views on these catalog tables. Then retry the failing CATMAINT *operation*. This message is followed by an abend04E RC00E40601. See the problem determination text of RC00E40601 for additional information.

DSNU777I *csect-name* **CATMAINT** *operation* **STATUS** *status-msg*

Explanation: The CATMAINT utility is processing CATMAINT *operation*. *status-msg* indicates where CATMAINT is in the *operation* process.

Severity: 0 (informational)

System action: Utility processing continues normally.

DSNU778I *csect-name* **- ERROR PROCESSING SQL**
STATEMENT - SQL CODE IS: *sql-code*
SQL MESSAGE TEXT: *message-text* **SQL**
STATEMENT: *sql-statement*

Explanation: A utility terminated abnormally in the
DB2 database services address space (ssnmDBM1)
during CATMAINT UPDATE processing. The
message-text indicates the problem with the SQL
statement. The problem SQL statement *sql-statement* is
also displayed.

Severity: 0 (informational)

System action: Utility processing terminates
abnormally.

User response: This message is followed by an
abend04E RC00E40601. The *sql-code*, *message-text*, and
sql-statement listed in this message are probably
sufficient to diagnose and correct the problem. See the
problem determination text of RC00E40601 for
additional information.

DSNU780I *csect-name* **DSNUTILB LEVEL** *rel1* **INVALID WITH SUBSYSTEM LEVEL** *rel2*

Explanation: JCL has been submitted which executes level 'rel1' of DSNUTILB against a level 'rel2' DB2 subsystem. The levels must match.

Severity: 8 (error)

System action: The utility terminates with a return code of 8.

User response: Determine the DB2 subsystem against which you wish to execute. Determine which STEPLIB data set contains the corresponding DSNUTILB load module. Correct the JCL to specify the correct data set on the STEPLIB DD statement and the correct DB2 subsystem on the EXEC statement and resubmit the job.

DSNU781I *csect-name* **RESTART OF LEVEL** *rel1* **UTILITY AT LEVEL** *rel2* **INVALID**

Explanation: JCL has been submitted which attempts to RESTART a stopped utility. The utility began processing at level 'rel1' and cannot be restarted at level 'rel2'.

System action: The utility job is abended with reason code X'00E40018'.

User response: Either restart the utility at the same level it was originated or terminate the utility with the TERM command and resubmit it at the new level. Refer to *DB2 Command Reference* for special TERM command considerations.

DSNU782I *csect-name* **UNABLE TO RECOVER DUE TO ACTIVITY AT ANOTHER SYSTEM LEVEL**

Explanation: The RECOVER utility is unable to recover the specified table space. Activity at another system level has left all image copies for this table space inconsistent with the DB2 LOG.

Severity: 8 (error)

System action: The utility terminates with a return code of 8.

User response: The table space, in its entirety can not be recovered. Individual data sets may be recoverable using the DSNUM() keyword.

DSNU783I *csect-name* **INDEXES DSNSCT02, DSNLUX01, DSNLUX02 and DSNLLX02 HAVE SUBPAGES VALUE GREATER THAN 1 AND CANNOT BECOME GROUP BUFFER POOL DEPENDENT.**

Explanation: CATMAINT CONVERT TO TYPE 1 DIRECTORY alters the indexes DSNSCT02, DSNLUX01, DSNLUX02, and DSNLLX02 to type 1 with more than one subpage. In a data sharing environment, there cannot be inter-DB2 R/W interest on type 1 indexes with more than one subpage (they cannot become *group buffer pool dependent*). This means that any read or update request that causes inter-DB2 R/W interest on these directory indexes is denied with a 'resource unavailable' condition.

For directory indexes, only type 2 indexes can have full R/W sharing among data sharing members.

Severity: 4 (warning)

System action: Processing continues.

User response: With very few exceptions, indexes on the directory will have inter-DB2 R/W interest if there are two or more members in your DB2 data sharing group. Use CATMAINT CONVERT TO TYPE 2 DIRECTORY to convert the indexes to type 2 if you plan to have two or more DB2 members.

DSNU784I *csect-name* **CATMAINT** *operation* **- CATALOG CONTAINS UNSUPPORTED OBJECTS:** *error-text* **OBJECT NAME -** *object-name*, **CREATOR -** *creator*

Explanation: The CATMAINT utility halted because the catalog contains an unsupported object/item (indicated by *error-text*):

- CATALOG CONTAINS TYPE 1 INDEXES

Severity: 8 (error)

System action: Utility processing halts. No SYSABEND dump or SVC dump is requested.

User response: Any unsupported objects/items should be removed from the DB2 catalog in the previous release. The DSNTIJPM and DSNTESQ jobs can be used to identify any unsupported objects that are found in a subsystem. Remove any unsupported objects in the DB2 catalog and retry the failing CATMAINT *operation*. This message is followed by an abend 04E RC 00E40601.

DSNU790I *csect-name utility-name option* **IS NOT ALLOWED WHEN THERE ARE OUTSTANDING UTILITIES STARTED FROM PRIOR RELEASES.**

Explanation: The utility to be executed may have conflicts with other outstanding utilities started from prior releases.

utility-name is the name of the utility that detected the conflict.

option is the option specified to the utility that detected the conflict.

System action: The request is rejected.

User response: Use the DISPLAY UTILITY command to see what utilities are outstanding. Issue the TERM UTILITY command on the same release as the release on which the outstanding utility was started.

DSNU795I *csect-name* **- CHECKLOB PHASE COMPLETE - ELAPSED TIME=***time*

Explanation: The CHECK LOB utility CHECKLOB phase was completed with the elapsed time of *time*.

Severity: 0 (informational)

System action: The CHECKLOB phase is completed.

DSNU796I *csect-name* **- REPRTLOB PHASE COMPLETE - ELAPSED TIME=***time*

Explanation: The CHECK LOB utility REPRTLOB phase was completed with the elapsed time of *time*.

Severity: 0 (informational)

System action: The REPRTLOB phase is completed.

DSNU805I *csect-name* **- TABLE=***table_name* **COLUMN=***column_name* **IS OUT OF SYNCH ROWID=***rowid* **VERSION=***version*

Explanation: The CHECK DATA utility found an auxiliary index entry for the LOB column named *column_name* with ROWID, *rowid*, and version, *version*. The base table row with ROWID, *rowid*, has a version value greater then that of the index record for the column specified.

System action: If **AUXERROR REPORT** was specified, this error is reported only. The base table space is set to an Aux Check Pending state.

If **AUXERROR INVALIDATE** was specified, the LOB column is set invalid. The base table space is set to an Aux Warning state.

Severity: 4 (Warning)

Programmer response: CHECK DATA relies on information in LOB table space and in the auxiliary index being correct. Before running CHECK DATA to check auxiliary columns, first run CHECK LOB and then run REBUILD INDEX or CHECK INDEX. CHECK LOB ensures the validity of the LOB table space. REBUILD INDEX or CHECK INDEX ensures the validity of the auxiliary index.

If **AUXERROR REPORT** was specified, run the CHECK DATA utility with **AUXERROR INVALIDATE** to set the base table column invalid.

If **AUXERROR INVALIDATE** was specified, use SQL to update the LOB column or delete the base table row. Use REPAIR to delete the LOB from the LOB table space using the ROWID and version reported in the message. Rerun the CHECK DATA utility with **AUXERROR INVALIDATE** specified. If all Auxiliary Column Check errors were corrected, Aux Warning is reset.

The REPAIR utility can be used to reset the Aux Warning state.

DSNU806I *csect-name* **- TABLE=***table_name* **COLUMN=***column_name* **WAS SET INVALID ROWID=***rowid* **VERSION=***version*

Explanation: The CHECK DATA utility with **AUXERROR INVALIDATE** specified, set the LOB column named *column_name* invalid. The LOB column was in the row with ROWID, *rowid* in the table named *table_name*. The version for the LOB column was *version*.

System action: Utility returns completion code=4. The base table space is set to a Aux Warning state.

Programmer response: None

Use SQL to update the LOB column or delete the base table row. Rerun the CHECK DATA utility with

AUXERROR INVALIDATE specified. If all Auxilary Column Check errors were corrected, Aux Warning will be reset.

The REPAIR utility can be used to reset the Aux Warning state.

DSNU807I *csect-name* **- TABLE=***table_name* **COLUMN=***column_name* **WAS SET VALID ROWID=***rowid* **VERSION=***version*

Explanation: The CHECK DATA utility with **AUXERROR INVALIDATE** specified, set the LOB column named *column_name* valid. The LOB column was in the row with ROWID, *rowid* in the table named *table_name*. The version for the LOB column was *version*.

System action: Utility returns completion code=4.

Programmer response: None. If all Auxilary Column Check errors were corrected, Aux Warning will be reset.

DSNU809I *csect-name* **- TABLE=***table_name* **COLUMN=***column_name* **IS MISSING IN INDEX** *index-name* **ROWID=***rowid* **VERSION=***version*

Explanation: The CHECK DATA utility found a record in the table named *table-name* in the row with a ROWID of *rowid*. The column named *column-name* did not have an index entry in the on the auxiliary index named *index-name*. The version for the LOB column was *version*.

System action: If **AUXERROR REPORT** was specified, this error is reported only. The base table space is set to an Aux Check Pending state.

If **AUXERROR INVALIDATE** was specified, the LOB column is set invalid. The base table space is set to an Aux Warning state.

Utility returns completion code=4.

Programmer response: CHECK DATA relies on information in LOB table space and in the auxiliary index being correct. Before running CHECK DATA to check auxiliary columns, first run CHECK LOB and then run REBUILD INDEX or CHECK INDEX. CHECK LOB ensures the validity of the LOB table space. REBUILD INDEX or CHECK INDEX ensures the validity of the auxiliary index.

If **AUXERROR REPORT** was specified, run the CHECK DATA utility with **AUXERROR INVALIDATE** to set the base table column invalid.

If **AUXERROR INVALIDATE** was specified, use SQL to update the LOB column or delete the base table row. Rerun the CHECK DATA utility with **AUXERROR INVALIDATE** specified. If all Auxiliary Column Check errors were corrected, Aux Warning will be reset.

The REPAIR utility can be used to reset the Aux Warning state.

DSNU811I *csect-name* **- TABLE=***table_name* **COLUMN=***column_name* **IS INVALID ROWID=***rowid* **VERSION=***version*

Explanation: The CHECK DATA utility found a record in the table named *table-name* in the row with ROWID *rowid*. The column named *column-name* was marked invalid by a prior execution of the CHECK DATA utility with **AUXERROR INVALIDATE** specified. The version for the LOB column was *version*.

System action: Utility returns completion code=4. The base table space is set to a Aux Warning state.

Programmer response: If **AUXERROR INVALIDATE** was specified, use SQL to update the LOB column or delete the base table row. Rerun the CHECK DATA utility with **AUXERROR INVALIDATE** specified. If all Auxiliary Column Check errors were corrected, Aux Warning will be reset.

The REPAIR utility can be used to reset the Aux Warning state.

DSNU813I *csect-name* **- LOB IN TABLE SPACE=***lob_table_space_name* **WITH ROWID=***rowid* **AND VERSION=***version* **HAS NO BASE TABLE ROW**

Explanation: The CHECK DATA utility found a LOB in the LOB table space named *lob_table_space_name* but did not find a row in the base table for this LOB. The LOB has a ROWID of *rowid* and version of *version*.

System action: Utility returns completion code=4.

Programmer response: CHECK DATA relies on information in the index on the auxiliary table and LOB table space being correct. Before running CHECK DATA to check auxiliary columns, first run CHECK LOB and then run REBUILD INDEX or CHECK INDEX. CHECK LOB ensures the validity of the LOB table space. REBUILD INDEX or CHECK INDEX ensures the validity of the auxiliary index.

The LOB can be deleted using the REPAIR utility. Specify LOCATE ROWID VERSION DELETE.

DSNU814I *csect-name* **- TABLE=***table_name* **COLUMN=***column_name* **IS ZERO OR NULL BUT ENTRY IN INDEX** *index-name* **FOUND ROWID=***rowid* **VERSION=***version*

Explanation: The CHECK DATA utility found a record in the table named *table-name* in the row with a ROWID of *rowid*. The column named *column-name* was indicated null or zero length, but an index entry with the same ROWID was found in the index on the auxiliary table named *index-name*. The version for the LOB column was *version*.

System action: The fact that a LOB column entry exists in the index on the auxiliary table when the LOB column is marked null or has a null length does not hinder LOB column processing.

If **AUXERROR INVALIDATE** was specified, the LOB column is not set invalid for this error.

Severity: 4 (warning)

Programmer response: CHECK DATA relies on information in the index on the auxiliary table and LOB table space being correct. If you did not run CHECK LOB followed by REBUILD INDEX or CHECK INDEX prior to running CHECK DATA, then before running CHECK DATA again, run CHECK LOB followed by REBUILD INDEX or CHECK INDEX. CHECK LOB checks the validity of the LOB table space and REBUILD INDEX or CHECK INDEX ensures that the auxiliary index is valid.

The LOB can be deleted from the LOB table space by using **REPAIR LOCATE TABLESPACE ROWID DELETE**

DSNU815I *csect-name* **- TABLESPACE** *table-space-name* **IS IN AUX CHECK PENDING STATE**

Explanation: The CHECK DATA utility found *auxiliary check* errors. *AUXERROR REPORT* was specified, so errors were left as found.

Auxiliary check pending is a prohibitive state.

Previous error messages report the errors found.

System action: Utility returns completion code=4.

Programmer response: Use the CHECK DATA with *AUXERROR INVALIDATE*. This will modify the *auxiliary check* errors so they can be corrected and set the state of the table space to *auxiliary warning*.

After running CHECK DATA with *AUXERROR INVALIDATE*, use SQL to update the LOB column or delete the base table row. Rerun the CHECK DATA utility with *AUXERROR INVALIDATE* specified. If all *auxiliary check* errors were corrected, *auxiliary warning* will be reset.

The *auxiliary check pending* state can also be reset using REPAIR with *SET NOAUXCHKP*. Resetting *auxiliary check pending* with REPAIR will have unpredictable results.

DSNU816I *csect-name* **- TABLESPACE** *table-space-name* **IS IN AUX WARNING STATE**

Explanation: The CHECK DATA utility found *auxiliary check* errors. *AUXERROR INVALIDATE* was specified, so LOB columns in error were marked invalid.

Auxiliary warning is a non-prohibitive state. Updating the LOB column or deleting the row containing the LOB column by SQL is permitted. Valid LOBs may be accessed, however any attempt to retrieve an invalid modified LOB column by SQL will result in a -904 return code.

Previous error messages report the errors found.

Severity: 0 (informational)

System action: Processing continues.

Programmer response: Use SQL to update or delete the LOB columns in error. Use CHECK DATA with AUXERROR INVALIDATE to verify all errors were corrected and to reset the *auxiliary warning* state.

The *auxiliary warning* state can also be reset using REPAIR with *SET NOAUXWARN*.

DSNU830I *csect-name* **- INDEX=** *index-name* **IS IN REBUILD PENDING STATE**

Explanation: If the RECOVER utility issued this message, the corresponding table space to this index was recovered to a point in time with keyword TOCOPY, TOLOGPOINT, or TORBA. Thus, the index was placed in rebuild pending status.

If the REBUILD INDEX utility issued this message, the index is in rebuild pending status. Either duplicate index entries existed during the BUILD phase of the REBUILD INDEX, or invalid KEY/RIDs, which do not belong to the partition of the cluster index, were detected. The duplicate index entries or invalid KEY/RID entries must be resolved before starting the REBUILD INDEX utility again to rebuild the index and remove the index from rebuild pending status.

If the REORG utility with SHRLEVEL REFERENCE or CHANGE (or -TERM UTILITY) issued this message, the index is in rebuild pending status. During the renaming of a data set during REORG, either a data set with the source name did not yet exist or a data set with the target name already existed.

Severity: 4 (warning) or 8 (error)

User response: If the RECOVER or REORG utility (or -TERM UTILITY) issued this message, use the REBUILD INDEX utility to rebuild this index.

If the REBUILD INDEX utility issued this message resolve the duplicate index entries or invalid keys and rerun the REBUILD INDEX utility.

DSNU831I *csect-name* **ALL INDEXES OF TABLESPACE=** *table space-name* **ARE IN REBUILD PENDING**

Explanation: The RECOVER TORBA, TOLOGPOINT, or TOCOPY utility issues this message to inform the user of the state of all indexes belonging to the recovered table space. All indexes were put in rebuild pending status to prevent data and index inconsistency. All indexes should be rebuilt before they are used.

System action: The RECOVER utility completed successfully.

Severity: 4 (warning)

User response: Use the REBUILD utility with the ALL option to rebuild all indexes over the recovered table space.

DSNU832I *csect-name* **- INDEX=** *creator.index-name* **PART=** *partition number* **IS IN REBUILD PENDING STATE**

Explanation: If the RECOVER utility issued this message, one partition of a partitioned table space is being recovered to a point in time with keyword TOCOPY, TOLOGPOINT, or TORBA. The corresponding index for that partition was placed in rebuild pending status.

If the REBUILD INDEX utility issued this message, the index or the partition of a cluster index is in rebuild pending status. Either duplicate index entries existed during the BUILD phase of the REBUILD INDEX or INDEX PART, or invalid KEY/RIDs, which do not belong to the partition of the cluster index, were detected. The duplicate index entries or invalid KEY/RID entries must be resolved before starting the REBUILD INDEX utility again to rebuild the index and remove the index from rebuild pending status.

If the REORG utility with SHRLEVEL REFERENCE or CHANGE (or -TERM UTILITY) issued this message, one partition of a partitioned index was placed in rebuild pending status. During the renaming of a data set during REORG, either a data set with the source name did not yet exist or a data set with the target name already existed.

Severity: 4 (warning) or 8 (error)

User response: If the RECOVER or REORG utility (or -TERM UTILITY) issued the message, rebuild this index with the REBUILD INDEX utility.

If the REBUILD INDEX utility issued this message resolve duplicate index entries or invalid keys and rerun the REBUILD INDEX utility.

DSNU833I *csect-name* **- UNABLE TO RECOVER** *object_type object_name* **DUE TO ALTER COLUMN**

Explanation: RECOVER utility is unable to recover the index or index space as requested because an ALTER COLUMN VARCHAR has been performed. The current column definition is no longer consistent with the image copy and log records which would be used to perform the requested recovery.

Severity: 8 (error)

User response: Use the REBUILD INDEX utility to rebuild the index from the tablespace.

DSNU834I *csect-name* **- UNABLE TO RECOVER** *object_type object_name* **DSNUM** *dsnum* **DUE TO ALTER COLUMN**

Explanation: RECOVER utility is unable to recover the index or index space partition as requested because an ALTER COLUMN VARCHAR has been performed. The current column definition is no longer consistent with the image copy and log records which would be used to perform the requested recovery.

Severity: 8 (error)

User response: Use the REBUILD INDEX utility to rebuild the index partition from the tablespace.

DSNU860I *csect-name* **DIAGNOSE UTILITY COMPLETE**

Explanation: This message is issued in response to the normal completion of the DIAGNOSE utility.

Severity: 0 (informational)

System action: This message indicates the normal completion of the DIAGNOSE utility. Processing continues to the next utility statement in the utility job step.

| **DSNU861I** **DISPLAY MEPL FOR SYSTEM** *name*

Explanation: This message is the header message for output of the DIAGNOSE DISPLAY MEPL function. The contents of the MEPL is dumped to SYSPRINT.

| The contents of the MEPL for subsystem *name* is
| dumped to SYSPRINT.

Severity: 0 (informational)

System action: This message indicates the beginning of DIAGNOSE DISPLAY MEPL processing.

DSNU862I **DISPLAY AVAILABLE UTILITIES. MAP:** *bit-map*

Explanation: This message is the end-result of a DIAGNOSE DISPLAY AVAILABLE control statement. It displays the utilities that are available on this subsystem. The message contains a machine-readable bit-map, and precedes a table of utility names. They are two representations of the same information. An available utility is represented by the utility name that appears in the table and a ″1″ in the corresponding bit position in the bit-map. An unavailable utility is represented by a blank table entry and a ″0″ in the corresponding bit position in the bit-map.

Utilities may be unavailable if separately-priced
features JDB881K and/or JDB881M are not installed, or
installed improperly.

The table position and bit position are fixed for any given utility, and maintenance cannot alter them. Bit positions correspond to table entries when you read the

table across each row, left to right, starting with the top row. For example, bit 1 corresponds to CATMAINT, bit 2 to CHECK, and bit 9 to OPTIONS.

The following example shows the partial output of this message. Due to space constraints, the sample has been truncated. The actual table contains eight columns of identity names:

```
DSNU862I  DSNUDIAG - DISPLAY AVAILABLE UTILITIES.
  MAP: 11111111111111111110000000000000
---------------------------------------------
|CATMAINT |CHECK    |COPY     |DIAGNOSE |...
|OPTIONS  |QUIESCE  |REBUILD  |RECOVER  |...
|STOSPACE |TEMPLATE |UNLOAD   |         |...
|         |         |         |         |
---------------------------------------------
```

Severity: 0 (informational)

System action: The DIAGNOSE DISPLAY AVAILABLE utility control statement has completed normally.

DSNU863I *csect-name* **DISPLAY OBD** *dbname*.*tsname*

Explanation: This message is the heading message for DIAGNOSE DISPLAY OBD output.

Severity: 0 (informational)

System action: This message indicates the start of the output for the DIAGNOSE DISPLAY OBD function.

DSNU864I *csect-name* **DISPLAY OBD, DBID=***nn* **OBID=***nn* **OBD TYPE=***type*

Explanation: This message is the heading message for the OBD being displayed.

Severity: 0 (informational)

System action: This message identifies the OBD being displayed.

DSNU865I *csect-name* **DISPLAY OBD** *dbname*.*tsname* **COMPLETE**

Explanation: This message is issued in response to normal completion of DISPLAY OBD processing.

Severity: 0 (informational)

System action: This message indicates the end of DISPLAY OBD processing.

DSNU866I *csect-name* **DISPLAY SYSUTIL JOB INFORMATION**

Explanation: This message is the header message for the SYSUTIL JOB INFORMATION section of the DIAGNOSE DISPLAY SYSUTIL output. This section displays the job information area of the SYSUTIL record and contains information which identifies the utility the SYSUTIL record is referencing.

Severity: 0 (informational)

System action: This message identifies the SYSUTIL JOB INFORMATION section of the DIAGNOSE DISPLAY SYSUTIL output.

DSNU867I *csect-name* **DISPLAY SYSUTIL CHECKPOINT/RESTART INFORMATION**

Explanation: This message is the header message for the SYSUTIL UTILITY CHECKPOINT/RESTART INFORMATION section of the DIAGNOSE DISPLAY SYSUTIL output. This section displays the utility checkpoint/restart area of the SYSUTIL record and contains information pertaining to utility processing needed for the restart of the utility.

Severity: 0 (informational)

System action: This message identifies the SYSUTIL UTILITY CHECKPOINT/RESTART INFORMATION section of the DIAGNOSE DISPLAY SYSUTIL output.

DSNU868I *csect-name* **DISPLAY SYSUTIL UTILITY DEPENDENT AREA**

Explanation: This message is the header message for the SYSUTIL UTILITY DEPENDENT INFORMATION section of the DIAGNOSE DISPLAY SYSUTIL output. This section displays the utility dependent area of the SYSUTIL record and contains information particular to the utility which the SYSUTIL record is referencing.

Severity: 0 (informational)

System action: This message identifies the SYSUTIL UTILITY DEPENDENT INFORMATION section of the DIAGNOSE DISPLAY SYSUTIL output.

DSNU869I *csect-name* **DISPLAY SYSUTIL COMPLETE**

Explanation: This message is issued in response to normal completion of DISPLAY SYSUTIL processing.

Severity: 0 (informational)

System action: This message indicates the end of DISPLAY SYSUTIL processing.

DSNU870I *csect-name* **NO SYSUTIL RECORD FOUND**

Explanation: This message is issued if no SYSUTIL records are found when DISPLAY SYSUTIL is specified.

Severity: 4 (warning)

System action: This message indicates the end of DISPLAY SYSUTIL processing.

DSNU871I *csectname* **DISPLAY DBET DATABASE** *dbname*

Explanation: This message is the heading for display output of the database exception table (DBET) entry associated with database *dbname*.

Severity: 0 (informational)

System action: DIAGNOSE DISPLAY DBET processing started for the named database.

DSNU872I *csectname* **DISPLAY DBET TABLE SPACE** *tsname*

Explanation: This message is the heading for display output of the database exception table (DBET) entry associated with table space *tsname*.

Severity: 0 (informational)

System action: DIAGNOSE DISPLAY DBET processing started for the named table space.

DSNU873I *csectname* **DISPLAY DBET INDEX** *index-name*

Explanation: This message is the heading for display output of the database exception table (DBET) entry associated with index *index-name*.

Severity: 0 (informational)

System action: DIAGNOSE DISPLAY DBET processing started for the named index.

DSNU874I *csectname* **NO DBET ENTRY FOUND FOR THE SPECIFIED OBJECT**

Explanation: There is no database exception table (DBET) entry associated with the specified object.

Severity: 4 (warning)

System action: Processing continues normally.

DSNU875I *csectname* **DIAGNOSE WAIT AT MESSAGE=***msgno* **INSTANCE=***nn* **CORR ID=***id1* **CONN ID=***id2* **REPLY TO CONT**

Explanation: The DIAGNOSE WAIT function was invoked to wait at a specified message 'msgno' on the nn'th time that message was issued. The CORR ID is the correlation identifier of the thread associated with the waiting application. The CONN ID is the connection identifier of the thread.

Severity: 4 (warning)

System action: Processing is stopped until the user responds to this message.

User response: See operator response.

Operator response: After diagnosis procedure in progress is completed, reply to this message to continue execution. Any reply is valid.

DSNU876I *csectname* **DIAGNOSE WAIT AT TRACEID=X'trcid' INSTANCE=***nn* **CORR ID=***id1* **CONN ID=***id2* **REPLY TO CONT**

Explanation: The DIAGNOSE WAIT function was invoked to wait at a specified traceid 'trcid' on the nn'th time that traceid was encountered. The CORR ID is the correlation identifier of the thread associated with the waiting application. The CONN ID is the connection identifier of the thread.

Severity: 4 (warning)

System action: Processing is stopped until the user responds to this message.

User response: See operator response.

Operator response: After diagnosis procedure in progress is complete, reply to this message to continue execution. Any reply is valid.

DSNU877I event modulename+offset

Explanation: Event 'event' occurred at module 'modulename' plus 'offset'. The DIAGNOSE utility issues this message to aid in diagnosing utility problems.

Severity: 0 (informational)

System action: Processing continues normally.

DSNU900I *csect-name* **- DBD X'xxxx' COMPARISON STARTING**

Explanation: This message marks the start of a DBD comparison for the REPAIR DBD TEST or REPAIR DBD DIAGNOSE utility. The DBID of the DBD being compared against the DB2 Catalog is 'xxxx'.

Severity: 0 (informational)

System action: DBD checking continues normally.

DSNU901I *csect-name* **- OBD X'xxxx'** *obd-type* **HAS THE FOLLOWING DIFFERENCES**

Explanation: This message marks the start of an OBD comparison for the REPAIR DBD TEST or REPAIR DBD DIAGNOSE utility. This message is displayed when a difference in the OBDs has been detected. The OBID of the OBD being compared against the DB2 catalog is 'xxxx'. The type of OBD being displayed is 'obd-type'.

Severity: 0 (informational)

System action: DBD checking continues normally.

DSNU902I *csect-name* **- EXISTING OBD AT OFFSET X'xxxxxxxx' IN EXISTING DBD**

Explanation: Following this message is a hexadecimal dump of the OBD as it exists in the DSNDB01.DBD01 page set. In the message, 'xxxxxxxx' is the hexadecimal offset from the beginning of the existing DBD as it appears later in the REPAIR DBD DIAGNOSE output.

Severity: 0 (informational)

System action: DBD checking continues normally.

DSNU903I *csect-name* **- REBUILT OBD AT OFFSET X'xxxxxxxx' IN REBUILT DBD**

Explanation: Following this message is a hexadecimal dump of the OBD that was rebuilt from the DB2 Catalog. In the message, 'xxxxxxxx' is the hexadecimal offset from the beginning of the rebuilt DBD as it appears later in the REPAIR DBD DIAGNOSE output.

Severity: 0 (informational)

System action: DBD checking continues normally.

DSNU904I *csect-name* **- EXISTING VALUE X'xx..xx'**

Explanation: This message shows a hexadecimal string which is the value of a field in the OBD as it exists on the DSNDB01.DBD01 page set. The hexadecimal string is 'xx..xx'.

Severity: 0 (informational)

System action: DBD checking continues normally.

DSNU905I *csect-name* **- REBUILT VALUE X'xx..xx'**

Explanation: This message shows a hexadecimal string which is the value of a field in the OBD which was rebuilt from the DB2 catalog. The hexadecimal string is 'xx..xx'.

Severity: 0 (informational)

System action: DBD checking continues normally.

DSNU906I *csect-name* **- EXISTING VALUE 'cc..cc'**

Explanation: This message shows a character string which is the value of a field in the OBD as it exists on the DSNDB01.DBD01 page set. The character string is 'cc..cc'.

Severity: 0 (informational)

System action: DBD checking continues normally.

DSNU907I *csect-name* **- REBUILT VALUE 'cc..cc'**

Explanation: This message shows a character string which is the value of a field in the OBD which was rebuilt from the DB2 catalog. The character string is 'cc..cc'.

Severity: 0 (informational)

System action: DBD checking continues normally.

DSNU908I *csect-name* **- EXISTING OBD IS** *obd-type*

Explanation: This message shows the OBD type of the OBD which exists on the DSNDB01.DBD01 page set.

Severity: 0 (informational)

System action: DBD checking continues normally.

DSNU909I *csect-name* **- REBUILT OBD IS** *obd-type*

Explanation: This message shows the OBD type of the OBD which was rebuilt from the DB2 catalog.

Severity: 0 (informational)

System action: DBD checking continues normally.

DSNU910I *csect-name* **- EXISTING DBD**

Explanation: Following this message is a hexadecimal dump of the DBD which exists on the DSNDB01.DBD01 page set.

Severity: 0 (informational)

System action: DBD checking continues normally.

DSNU911I *csect-name* **- REBUILT DBD**

Explanation: Following this message is a hexadecimal dump of the DBD which was rebuilt from the DB2 Catalog.

Severity: 0 (informational)

System action: DBD checking continues normally.

DSNU912I *csect-name* **-** *phase-completion-message* **--** *nn* **ERRORS WERE FOUND**

Explanation: This message marks the end of a processing phase in the DBD comparison for a REPAIR DBD TEST or REPAIR DBD DIAGNOSE utility. In the message, 'nn' indicates the number of errors detected while processing. Informational and warning messages are not added into the error count.

Severity: 0 (informational)

System action: DBD checking continues normally.

DSNU913I *csect-name* **- OBD X'xxxx' COMPARISON COMPLETE --** *nn* **ERRORS WERE FOUND**

Explanation: This message marks the end of an OBD comparison for the REPAIR DBD TEST or REPAIR DBD DIAGNOSE utility. This message is displayed only when a difference has been detected. If there is no difference, this message will not be displayed. The OBID of the OBD being compared against the DB2

catalog is 'xxxx'. The number of errors in this OBD is 'nn'. Informational and warning messages are not added into the error count.

Severity: 0 (informational)

System action: DBD checking continues normally.

DSNU914I *csect-name* **- DBD X'xxxx' COMPARISON COMPLETE -- *nn* ERRORS WERE FOUND**

Explanation: This message marks the end of a DBD comparison for the REPAIR DBD TEST or REPAIR DBD DIAGNOSE utility. The DBID of the DBD being compared against the DB2 catalog is 'xxxx'. The total number of errors found in this DBD is 'nn'.

Severity: 0 (informational)

System action: DBD checking completed.

DSNU915I *csect-name* **- OBD X'obid' E-X'xxxxxxxx' R-X'yyyyyyyy'** *obd-description*

Explanation: This message is used when listing the OBDs in the REPAIR DBD DIAGNOSE utility. The OBID of the OBD being listed is 'obid'. The offset to the OBD within the EXISTING DBD which is displayed later in the output is 'xxxxxxxx'. The offset to the OBD within the REBUILT DBD which is displayed later in the output is 'yyyyyyyy'. A short description of the state of the OBD is 'obd-description'.

Severity: 0 (informational)

System action: DBD checking completed.

DSNU916I *csect-name* **- OBDS DIFFER AT OFFSET X'xxxxxxxx'** *field-name* **- INFORMATION NOT IN DB2 CATALOG**

Explanation: This message indicates the value of a field in the OBD differs from the DBD which exists on the DSNDB01.DBD01 page set and the DBD rebuilt from the DB2 catalog. This is information which is not kept in the DB2 catalog. This is an acceptable difference and does not indicate that the OBD is damaged. The hexadecimal offset from the beginning of the OBD is 'xxxxxxxx'. The name of the field is 'field-name'.

Severity: 0 (informational)

System action: DBD checking continues normally.

DSNU917I *csect-name* **- DBDS DIFFER AT OFFSET X'xxxxxxxx'** *field-name* **- INFORMATION NOT IN DB2 CATALOG**

Explanation: This message indicates that the value of a field in the DBD differs from the DBD which exists on the DSNDB01.DBD01 page set and the DBD rebuilt from the DB2 Catalog. This is information which is not kept in the DB2 catalog. This is an acceptable difference and does not indicate that the DBD is damaged. The hexadecimal offset from the beginning of the DBD is 'xxxxxxxx'. The name of the field is 'field-name'.

Severity: 0 (informational)

System action: DBD checking continues normally.

DSNU919I *csect-name* **- DBD X'xxxx' OBD X'yyyy' -** *details*

Explanation: This message gives detailed information
about the specified OBD. ′xxxx′ is the DBID of the
DBD. ′yyyy′ is the OBID of the OBD or the PSID of the
OBD when the *details* indicates a PSID is being
processed. *details* provides additional information.

Severity: 0 (informational)

System action: DBD checking continues normally.

DSNU920I *csect-name* **- OBDS DIFFER AT OFFSET X'xxxxxxxx'** *field-name*

Explanation: This message indicates that the value of a field in the OBD differs from the DBD which exists on the DSNDB01.DBD01 page set and the DBD rebuilt from the DB2 catalog. The DB2 Catalog and DBD/OBD are inconsistent. The hexadecimal offset from the beginning of the OBD is 'xxxxxxxx'. The name of the field is 'field-name'.

Severity: 4 (warning)

System action: DBD checking continues normally.

DSNU921I *csect-name* **- OBD X'xxxx' MISSING IN EXISTING DBD**

Explanation: This message indicates that an OBD which was rebuilt from the DB2 catalog does not exist in the DBD which exists on the DSNDB01.DBD01 page set. The DB2 catalog and DBD/OBD are inconsistent. The OBID of the missing OBD is 'xxxx'.

Severity: 4 (warning)

System action: DBD checking continues normally.

DSNU922I *csect-name* **- OBD X'xxxx' MISSING IN REBUILT DBD**

Explanation: This message indicates that an OBD in the DBD which exists on the DSNDB01.DBD01 page set does not exist in the DBD which was rebuilt from the DB2 catalog. The DB2 catalog and DBD/OBD are inconsistent. The OBID of the missing OBD is 'xxxx'.

Severity: 4 (warning)

System action: DBD checking continues normally.

DSNU923I *csect-name*—**OBDTYPE FOR OBD X'xxxx' DOES NOT MATCH —NO FURTHER CHECKING DONE FOR THIS OBD**

Explanation: The object descriptor (OBD) that exists on the DSNDB01.DBD01 page set is different from the type of OBD that was rebuilt from the DB2 catalog. No further processing can be done because the fields of one OBD type are different from the fields of another. The DB2 catalog and database descriptor (DBD)/OBD are inconsistent. The object identifier (OBID) of the OBD that is inconsistent is 'xxxx'.

Severity: 4 (warning)

System action: Checking for OBD X'xxxx' stops. Checking of the DBD continues normally.

DSNU924I *csect-name*—**DBDS DIFFER AT OFFSET X'xxxxxxxx'** *field-name*

Explanation: The value of a field in the database descriptor (DBD) differs from the value of the corresponding field in the DBD that exists on the DSNDB01.DBD01 page set and differs from the value of the corresponding field in the DBD that was rebuilt from the DB2 catalog.

xxxxxxxx
The hex offset from the beginning of the DBD.

field-name
The name of the field.

Severity: 0 (informational)

System action: Checking of the DBD continues normally.

DSNU925I *csect-name*—**OBDDFILE AT OFFSET X'xxxxxxxx' IN EXISTING DBD HAS OBID X'nnnn' WHICH IS NOT A FILE OBD**

Explanation: The field that points to the first FILE OBD (OBDDFILE) in the database descriptor (DBD) that exists on the DSNDB01.DBD01 page set does not point to a FILE OBD. The DB2 catalog and DBD/OBD (object descriptor) are inconsistent.

xxxxxxxx
The hex offset from the beginning of the existing DBD.

nnnn The object identifier (OBID) of the OBD which should be a FILE OBD.

Severity: 4 (warning)

System action: Checking of the DBD continues normally.

DSNU926I *csect-name*—**OBDDFILE AT OFFSET X'xxxxxxxx' IN REBUILT DBD HAS OBID X'nnnn' WHICH IS NOT A FILE OBD**

Explanation: The field that points to the first FILE OBD (OBDDFILE) in the database descriptor (DBD) that was rebuilt from the DB2 catalog does not point to a FILE OBD. The DB2 catalog and DBD/OBD (object descriptor) are inconsistent.

xxxxxxxx
The hex offset from the beginning of the rebuilt DBD.

nnnn The object identifier (OBID) of the OBD that should be a FILE OBD.

Severity: 4 (warning)

System action: Checking of the DBD continues normally.

DSNU927I *csect-name*—**OBD X'xxxx' IS A HOLE IN EXISTING DBD**

Explanation: The object descriptor (OBD) that exists on the DSNDB01.DBD01 page set has been logically deleted, OBDHOLE = '1'B. The DB2 catalog and database descriptor (DBD)/OBD are inconsistent. The object identifier (OBID) of the OBD that is a hole is 'xxxx'.

Severity: 4 (warning)

System action: Checking of the DBD continues normally.

DSNU928I *csect-name*—**OBD X'xxxx' IS A HOLE IN REBUILT DBD**

Explanation: The object descriptor (OBD) that was rebuilt from the DB2 catalog has been logically deleted, OBDHOLE = '1'B. The DB2 catalog and database descriptor (DBD)/OBD are inconsistent. The object identifier (OBID) of the OBD that is a hole is 'xxxx'.

Severity: 4 (warning)

System action: Checking of the DBD continues normally.

DSNU929I *csect-name*—**OBD FIELD** *field-name* **HAS VALUE OUT OF RANGE AT OFFSET X'xxxxxxxx' IN OBD X'nnnn'**

Explanation: The object descriptor (OBD) field 'field-name' has an unexpected value. The DB2 catalog and database descriptor (DBD)/OBD are inconsistent.

field-name
The name of the field in the OBD.

xxxxxxxx
The hex offset from the beginning of the OBD.

nnnn The object identifier (OBID) of the OBD.

Severity: 4 (warning)

System action: Checking of the DBD continues normally.

DSNU930I *csect-name*— **INDEX** *index-name* **(OBD X'xxxx') MUST BE REBUILT**

Explanation: The index 'index-name' must be either recovered or dropped and recreated to ensure that the index is consistent with the description in the INDEX FANSET OBD. The object descriptor (OBD) is not damaged. However, if the index is not either recovered or dropped and recreated, incorrect output can be produced when the index is used as an access path.

index-name
The name of the index found in SYSIBM.SYSINDEXES.

xxxx The object identifier (OBID) of the OBD.

Severity: 8 (error)

System action: Checking of the DBD continues normally.

DSNU931I *csect-name* **- OBDDMAX FIELD OF DBD IS TOO LARGE**

Explanation: The OBDDMAX field (maximum number of ID in the DBD) is too large. It is larger than the largest possible number of ID based upon the DBD size.

Severity: 8

System action: Checking of the DBD continues but using the largest possible number of ID instead of the incorrect OBDDMAX.

DSNU932I REPAIR CAN NOT DROP THE DATABASE.

Explanation: The system cannot drop the database because the name that was specified for the database and the DBID that was specified do not match.

System action: The utility job is halted.

User response: Correct the utility statement, and resubmit the job.

DSNU935I *csect-name* **DATABASE** *database-name* **IS NOT STARTED FOR UTILITY PROCESSING ONLY**

Explanation: This message is issued when the REPAIR utility with the DBD option is run on a database that is not started for utility processing only.

Severity: 8

System action: REPAIR DBD terminated.

User response: Stop the database. Then start the database with the ACCESS(UT) option. Resubmit the REPAIR DBD job.

DSNU936I *csect-name* **DATABASE** *database-name* **IS IN USE**

Explanation: This message is issued when the REPAIR utility with the DBD option is run on a database that is in use by others.

Severity: 8

System action: REPAIR DBD terminated.

User response: Wait until all users are finished, then resubmit the REPAIR DBD job.

DSNU938I *csect-name*—**OBD X'xxxx' HAD DIFFERENCES WHICH WERE NOT IN A NAMED FIELD**

Explanation: There is a difference between the object descriptor (OBD) in the database descriptor (DBD) that exists on the DSNDB01.DBD01 page set and the DBD that was rebuilt from the DB2 catalog. This difference does not exist in a named field. Therefore, the detailed search by field name fails to find the difference. The DB2 catalog and DBD/OBD are inconsistent. The object identifier (OBID) of the OBD in error is 'xxxx'.

Severity: 4 (warning)

System action: Checking of the DBD continues normally.

DSNU939I *csect-name*—**DBD HEADER HAD DIFFERENCES WHICH WERE NOT IN A NAMED FIELD**

Explanation: There is a difference between the DBD header in the DBD that exists on the DSNDB01.DBD01 page set and the DBD that was rebuilt from the DB2 catalog. This difference does not exist in a named field. Therefore, the detailed search by field name fails to find the difference. The DB2 catalog and database descriptor (DBD)/object descriptor (OBD) are inconsistent.

Severity: 4 (warning)

System action: Checking of the DBD continues normally.

DSNU940I *csect-name* **AUTHID OR LUNAME NOT BLANK FOR FOLLOWING SYSPROCEDURES ROW:**

Explanation: A SYSPROCEDURES row contains an AUTHID or LUNAME value. AUTHID and LUNAME are not supported attributes for DB2 Version 6 stored procedures. DSNU944I messages follow that show the contents of the SYSPROCEDURES row.

System action: The SYSPROCEDURES row is not migrated. SYSPROCEDURES migration continues with the next row.

System programmer response: Execute a CREATE

PROCEDURE statement to add the stored procedure definition. A schema name may be specified in place of AUTHID and LUNAME in the stored procedure definition.

DSNU941I *csect-name* **CREATE PROCEDURE FAILED WITH SQLCODE=***sqlcode* **SQLSTATE=***sqlstate*

Explanation: The migration of a SYSPROCEDURES row has failed. The SQLCODE and SQLSTATE values indicate the type of error that occurred.

System action: The SYSPROCEDURES row is not migrated. The SYSPROCEDURES migration continues with the next row.

System programmer response: Check the stored procedure attributes and parameters that are listed in the DSNU942I message, which follows this message.

Execute a CREATE PROCEDURE statement with correct attributes and parameters to add the stored procedure definition.

DSNU942I *csect-name create_procedure_statement*

Explanation: This message follows message DSNU941I when the migration of a SYSPROCEDURES row fails. This message contains the CREATE PROCEDURE statement that DB2 used for migration of the SYSPROCEDURES row.

System action: The SYSPROCEDURES row is not migrated. SYSPROCEDURES migration continues with the next row.

System programmer response: Use the CREATE PROCEDURE statement in this message and the SQLCODE and SQLSTATE information in message DSNU941I to diagnose the error.

DSNU944I *csect-name sysprocedures_column_name* **-** *sysprocedures_column_value*

Explanation: This message lists a column value for a SYSPROCEDURES row that could not be migrated. This message follows message DSNU940I.

System action: The SYSPROCEDURES row is not migrated. SYSPROCEDURES migration continues with the next row.

DSNU945I *csect-name* **CREATE SUCCESSFUL FOR PROCEDURE** *procedure_name*

Explanation: A SYSPROCEDURES row with a value of *procedure_name*. in column PROCEDURE has been successfully migrated.

System action: SYSPROCEDURES migration continues normally.

DSNU971I *csect-name* **TABLESPACE** *dbname.tsname* **IS CHECK PENDING OR AUX CHECK PENDING CORRELATION-ID** *id1* **CONNECTION-ID** *id2* **LUW-ID** *id3*

Explanation: This message is sent to the console when TABLESPACE *tsname* is set to CHECK pending or auxiliary CHECK pending state. The CORRELATION-ID *id1* is the correlation identifier of the thread requesting CHECK pending or auxiliary CHECK pending status. Correlation IDs beginning with '0nn', where *nn* is a number from 01 to 28, identify system agents. The CONNECTION-ID *id2* is the correlation identifier of the thread. The LUW-ID *id3* is the logical-unit-of-work identifier of the thread. If the LUW-ID is an '*', the thread has no LUW-ID. If it is not an '*', the value is a logical-unit-of-work-id that is assigned to the thread followed by an '=' sign and a token value, which can be used in place of logical-unit-of-work-id in any DB2 command that accepts logical-unit-of-work-id as input.

Severity: 4 (warning)

System action: Processing continues.

User response: Refer to Part 2 (Volume 1) of *DB2 Administration Guide* for a discussion of the CHECK pending operation.

DSNU972I *csect-name* **CATALOG UPDATING FAILURE WHILE SETTING OR RESETTING CHECK PENDING CORRELATION-ID** *id1* **CONNECTION-ID** *id2* **LUW ID=***luw-id*

Explanation: When the check pending state is set on or off, the updating of the catalog tables (SYSTABLESPACE, SYSTABLES, and SYSTABLEPART) fails. The CORRELATION-ID, *id1*, is the correlation identifier of the thread associated with the application requesting that the check pending state be set on or off. The CONNECTION-ID, *id2*, is the correlation identifier of the application. Correlation IDs beginning with '0nn', where 'nn' is a number from 01 to 28, identify system agents.

Severity: 0 (informational) or 4 (warning)

System action: Processing continues.

Problem determination: The utility being processed requested the setting of check pending either on or off. The updating of the STATUS column of SYSTABLESPACE, and of the CHECKFLAG and CHECKRID columns of SYSTABLES and SYSTABLEPART failed. Processing continues normally. However, those columns for the rows being updated now contain invalid values. Subsequent utility processing ignores these values, and the next setting or resetting of check pending for that table space corrects the error in one of the following ways:

- If the table space is set to check pending and the catalog fields are in error, the scope of check pending is escalated in the catalog. (The STATUS column of SYSTABLESPACE is set to 'P', and the CHECKFLAG and CHECKRID columns of SYSTABLES and SYSTABLEPART are set to 'C' and binary zeros.)
- If the table space is set to not check pending and the catalog fields are in error, the scope of check pending is set off in the catalog. (The STATUS column of SYSTABLESPACE is set to an 'A', and the CHECKFLAG and CHECKRID columns of SYSTABLES and SYSTABLEPART are set to blanks.)

After the catalog fields are corrected, the setting or resetting of check pending continues as specified.

Usually message DSNT500I or DSNT501I is also issued, because the probable cause of the error is RESOURCE UNAVAILABLE. If the reason for the failure was not RESOURCE UNAVAILABLE, an SVC dump is requested and processing continues normally. To determine the cause of the failure, refer to the dump listing and the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 5, 32.

DSNU973I *csect-name* **CHECK PENDING, AUX CHECK PENDING OR BOTH HAS BEEN RESET FOR TABLESPACE** *dbname.tsname* **CORRELATION-ID** *id1* **CONNECTION-ID** *id2* **LUW-ID** *id3*

Explanation: This message is sent to the console when CHECK pending, auxiliary CHECK pending or both states were reset for TABLESPACE *tsname*. The CORRELATION-ID *id1* is the correlation identifier of the thread requesting that the CHECK pending state, the auxiliary CHECK pending state or both states be reset. Correlation IDs beginning with '0nn', where *nn* is a number from 01 to 28, identify system agents. The CONNECTION-ID *id2* is the correlation identifier of the thread. The LUW-ID *id3* is the logical-unit-of-work identifier of the thread. If the LUW-ID is an '*', the thread has no LUW-ID. If it is not an '*', the value is a logical-unit-of-work-id that is assigned to the thread followed by an '=' sign and a token value, which can be used in place of the logical-unit-of-work-id in any DB2 command that accepts a logical-unit-of-work-id as input.

Severity: 0 (informational)

System action: Processing continues.

User response: Refer to Part 2 (Volume 1) of *DB2 Administration Guide* for a discussion about clearing the CHECK pending status.

DSNU993I *csect-name* **IMAGE COPY RESTORE TERMINATED**

Explanation: Catalog and directory initialization (image copy restore) did not complete successfully.

Severity: 8 (error)

System action: Processing terminates.

User response: Examine other error messages associated with this job.

DSNU994I *csect-name* **IMAGE COPY RESTORE COMPLETED SUCCESSFULLY**

Explanation: Catalog and directory initialization (image copy restore) completed successfully.

Severity: 0 (informational)

DSNU995I *csect-name* **VSAM PUT ERROR,RPLERREG =** *return-code*, **RPLERRCD =** *reason-code*

Explanation: VSAM PUT failed with the indicated *return-code* and *reason-code*.

Severity: 8 (error)

System action: Processing terminates.

User response: Notify system programmer.

System programmer response: Follow standard VSAM diagnostic procedure. Refer to *z/OS DFSMS: Macro Instructions for Data Sets*. See also *MVS/DFP V3R3 Diagnosis Guide* for more information about the VSAM procedures.

DSNU996I *csect-name* **VSAM OPEN ERROR, ACBERRFLG =** *error-code*

Explanation: VSAM OPEN failed with the indicated ACB *error-code*.

Severity: 8 (error) if error is greater than or equal to 128; otherwise, 4 (warning)

System action: Processing terminates if the error code is greater than or equal to 128; otherwise, processing continues normally.

User response: Notify the system programmer.

System programmer response: Follow standard VSAM diagnostic procedure. Refer to *z/OS DFSMS: Macro Instructions for Data Sets*. See also *MVS/DFP V3R3 Diagnosis Guide* for more information about the VSAM procedures.

DSNU997I *csect-name* **OUTPUT DATA SET NAME =** *data-set-name*

Explanation: The indicated *data-set-name* is used for output.

Severity: 0 (informational)

DSNU998I *csect-name* **INPUT DATA SET NAME =** *data-set-name*

Explanation: The indicated *data-set-name* is used for input.

Severity: 0 (informational)

DSNU999I *csect-name* **START OF IMAGE COPYRESTORE FOR JOB** *job-name step-name*

Explanation: The catalog and directory initialization (image copy restore) started.

Severity: 0 (informational)

DSNU1000I *csect-name* **- PROCESSING CONTROL STATEMENTS IN PREVIEW MODE**

Explanation: The PREVIEW option has been specified either on the OPTIONS control statement or as a JCL parameter. Control statements will be previewed in this execution for TEMPLATE and LISTDEF use but the utility control statements will not be executed. PREVIEW via JCL parameter will continue to the end of the job step. PREVIEW via OPTIONS PREVIEW will contine until the end of the job step or another OPTIONS control statement.

Severity: 0 (informational)

System action: Utility control statements are previewed but not executed.

User response: If the return code is 8, PREVIEW detected errors which must be corrected. If the return code is 4, PREVIEW detected potential problems, but the previewed statements are valid.

Examine the preview output shown in informational messages DSNU1010I, DSNU1020I, DSNU1021I, DSNU1022I, DSNU1023I and DSNU1025I, plus any error messages. Correct any statements that are in error and remove the PREVIEW option to execute the control statements.

DSNU1001I *csect-name* **- LISTDEF** *list-name* **CLAUSE IDENTIFIES NO NEW OBJECTS:** *clause-type object-type qualifier.name*

Explanation: An INCLUDE or EXCLUDE clause of a LISTDEF control statement returned no new objects. This means either that the INCLUDE or EXCLUDE clause evaluated to no objects at all, or that it did not return any objects which were not already included or excluded by previous INCLUDE or EXCLUDE clauses. In either case, the total number of objects covered by the LISTDEF did not change as a result of this clause.

Severity: 4 (warning)

System action: Utility processing continues.

User response: Correcting the clause in error will eliminate this message in the future.

DSNU1002I *csect-name* **- INVALID KEYWORD SPECIFIED ON CLAUSE:** *clause-type object-type qualifier.name*

Explanation: An INCLUDE or EXCLUDE clause of a LISTDEF control statement has an invalid keyword specified.

Catalog and directory objects may only be specified via their TABLESPACE or INDEXSPACE names. DATABASE, TABLE and INDEX may not be specified. Catalog and directory object names may not contain pattern matching characters. They must be included or excluded explicitly using their complete names. Those include or exclude clauses may not contain the keywords RI, BASS, LOB, ALL.

Severity: 8 (error)

System action: Utility processing stops.

User response: Correct the invalid LISTDEF clause and rerun the utility.

DSNU1003I *csect-name* **- INVALID LIST FOR** *utility-name* **UTILITY**

Explanation: The list name referenced on this control statement is not valid for the specified utility. The list either contains no objects or contains objects which cannot be processed by this specific utility.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Correct the LISTDEF in error, and resubmit the utility job. Refer to *DB2 Utility Guide and Reference* for the correct option specifications.

DSNU1004I *csect-name* **- DUPLICATE** *object-type object-name* **FOUND IN SYSIN** *dd-name*

Explanation: Duplicate LISTDEF or TEMPLATE statements (duplicate names) were detected in the SYSIN, listdef, or template dataset.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Eliminate the duplicate LISTDEF or TEMPLATE names and resubmit the utility job. Refer to *DB2 Utility Guide and Reference* for the correct option specifications.

DSNU1005I *csect-name* **- UTILITY** *name* **NOT ALLOWED IN** *dd-type* **DD** *dd-name*

Explanation: The utility parser detected a control statement other than TEMPLATE in the SYSTEMPL data set or LISTDEF in the SYSLISTD data set.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Remove the invalid control statements and resubmit the utility job. Refer to *DB2 Utility Guide and Reference* for the correct option specifications.

DSNU1006I *csect-name* **-** *object-type object-name* **NOT FOUND**

Explanation: A utility control statement used a LIST or TEMPLATE that was not found in SYSIN or either of the SYSTEMPL or SYSLISTD data sets. These DD names may vary if OPTIONS LISTDEFDD or OPTIONS TEMPLATEDD were specified.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Provide the missing LISTDEF or TEMPLATE definition in SYSIN prior to its use and resubmit the utility job. If the definition utility exists in another data set, provide the SYSTEMPL or SYSLISTD DD card for the data set. Refer to *DB2 Utility Guide and Reference* for the correct option specifications.

DSNU1007I *csect-name* **- DATE/TIME VALUES MAY CHANGE BEFORE EXECUTION**

Explanation: PREVIEW processing has detected DSN date/time variables on a TEMPLATE control statement being used. These values may change before the actual utility execution, resulting in different data set names.

Severity: 4 (warning)

System action: Preview processing continues normally.

DSNU1008I *csect-name* **- LISTDEF** *name* **CONTAINS NO OBJECTS**

Explanation: The indicated LISTDEF list definition expands to contain no objects.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Correct the LISTDEF definition and resubmit the utility job. Refer to *DB2 Utility Guide and Reference* for the correct option specifications.

DSNU1009I *csect-name* **- TEMPLATE** *name* **DSN=** *dataset-name*

Explanation: PREVIEW processing has expanded a referenced TEMPLATE data set name to the value shown. If the data set name contains date/time variables, the date/time values might change before actual utility execution.

Severity: 8 (error)

System action: PREVIEW processing continues.

User response: If the data set names shown are not acceptible, correct the TEMPLATE control statements, and resubmit the utility job. Refer to *DB2 Utility Guide and Reference* for the correct option specifications.

DSNU1011I WORKFILE NOT ALLOWED IN LISTDEF

Explanation: A workfile database cannot be used in a LISTDEF list because utilities do not process them.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Correct the LISTDEF in error, and resubmit the utility job. Refer to *DB2 Utility Guide and Reference* for the correct option specifications.

DSNU1012I *csect-name* **- LISTDEF MUST BEGIN WITH AN INCLUDE CLAUSE**

Explanation: A LISTDEF control statement begins with something other than an INCLUDE clause. The LISTDEF control statement must begin with an INCLUDE clause.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Correct the LISTDEF in error, and resubmit the utility job. Refer to *DB2 Utility Guide and Reference* for the correct option specifications.

DSNU1013I *csect-name* **-** *dd-type* **DD CARD** *dd-name* **NOT FOUND**

Explanation: An OPTIONS control statement specifies a DD name '*dd-name*' that does not exist in the JCL.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Correct the DD name on the OPTIONS statement or include the missing JCL DD card, and resubmit the utility job. Refer to *DB2 Utility Guide and Reference* for the correct option specifications.

DSNU1014I *csect-name* **- INDEXSPACE** *name* **NOT FOUND**

Explanation: The specified INDEXSPACE does not exist.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Correct the operand in error, and resubmit the utility job. Refer to *DB2 Utility Guide and Reference* for the correct option specifications.

DSNU1015I *csect-name* **- ERROR ALLOCATING DATA SET DSN=***name* **CODE=X'***reason-code***'**

Explanation: An error was returned by SVC 99 during TEMPLATE data set dynamic allocation. This message is followed by text which explains the reason-code shown.

Severity: 8 (error)

System action: Utility processing for the current object halts. If a list is being processed, utility processing of the next object will proceed or halt based on the OPTIONS ITEMERROR control statement. By default, execution will HALT.

User response: Correct the error preventing dynamic allocation, and resubmit the utility job. In some cases, TEMPLATE dynamic allocation may not be possible and a JCL DD card may be required. See DYNALLOC (SVC 99) text unit documentation for limitations.

DSNU1016I *csect-name* **- TEMPLATE DSN VARIABLE** *var-name* **HAS NO VALUE**

Explanation: A utility has attempted to build a data set name based on a TEMPLATE that specifies DSN variables. The variable shown has no value in this particular use. For example, **PART** when the object is not partitioned or **ICTYPE** when an image copy is not being allocated.

Severity: 8 (error)

System action: Utility processing for the current object halts. If a list is being processed, utility processing of the next object will proceed or halt based on the OPTIONS ITEMERROR control statement. By default, execution will HALT.

User response: Correct the TEMPLATE DSN specification to eliminate the invalid variable or use a different TEMPLATE, and resubmit the utility job. Refer to *DB2 Utility Guide and Reference* for the correct option specifications.

DSNU1017I *csect-name* **- TEMPLATE DSN VARIABLE** *var-name* **TRUNCATED TO** *value*

Explanation: While performing dynamic allocation of a TEMPLATE data set the DSN variable shown was truncated to the value shown.

Severity: 4 (warning)

System action: Utility processing continues.

DSNU1018I *csect-name* **- USER** *user-id* **NOT AUTHORIZED TO ACCESS CATALOG TABLE** *qualifier***.** *name*

Explanation: The user executing the utility does not have the required authority to access the catalog table shown. This access is required to expand the LISTDEF control statement being used.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: GRANT the required authority to access the DB2 catalog to the user executing the utility. Refer to the LISTDEF section of *DB2 Utility Guide and Reference* for valid authorities.

DSNU1019I *csect-name* **- INVALID TEMPLATE NAME** *template-name*

Explanation: A TEMPLATE control statement specifies an invalid TEMPLATE name. The names UTPRINT, SORTLIB and any name starting with SORTWK or SYS may not be used.

Severity: 8 (error)

System action: Utility processing terminates.

User response: Correct the TEMPLATE name in error, and resubmit the utility job. Refer to *DB2 Utility Guide and Reference* for the correct option specifications.

DSNU1020I *csect-name* **- EXPANDING LISTDEF** *list-name*

Explanation: PREVIEW processing is performing the catalog lookup necessary to expand the specified LISTDEF.

Severity: 0 (informational)

System action: PREVIEW processing continues.

DSNU1021I *csect-name* **- PROCESSING** *clause-type* **CLAUSE** *object-type qualifier.name*

Explanation: PREVIEW processing is expanding the specified clause of a LISTDEF. A DSNU1022I message follows this message to indicate how many objects are identified by each INCLUDE or EXCLUDE clause and message DSNU1023I to show the final expanded list..

Severity: 0 (informational)

System action: PREVIEW processing continues.

DSNU1022I *csect-name* **- CLAUSE IDENTIFIES** *n* **OBJECTS**

Explanation: PREVIEW processing has expanded one INCLUDE or EXCLUDE clause of a LISTDEF control statement. The clause expanded to the indicated *n* number of objects. The sum of the values shown on the DSNU1022I messages may not match the quantity shown on DSNU1023I due to INCLUDEs, EXCLUDEs and list overlap.

Severity: 0 (informational)

System action: PREVIEW processing continues.

DSNU1023I *csect-name* **- LISTDEF** *name* **CONTAINS** *n* **OBJECTS**

Explanation: PREVIEW processing has finished expanding a LISTDEF. The final list contains the indicated number of objects. The value shown may not match the sum of previous DSNU1022I quantities due to INCLUDEs, EXCLUDEs and list overlap.

Severity: 0 (informational)

System action: PREVIEW processing continues.

DSNU1024I *csect-name* **- OPTIONS EVENT WARNING HAS SET RETURN CODE TO** *return-code*

Explanation: Utility processing has ended with WARNING messages and a return code of 4. However, due to a previous OPTIONS EVENT WARNING control statement, the return code has been changed to the indicated return code.

Severity: 0 (informational) or 8 (warning)

System action: Utility processing halts if WARNING RC8 was specified, or continues if WARNING RC0 was specified.

User response: Use OPTIONS WARNING RC0 only on repetitive utility executions where WARNING messages are expected and acceptable.

DSNU1025I *csect-name* **-** *name* **IS A DD NAME**

Explanation: PREVIEW processing has checked the current utility control statement for TEMPLATE dynamic allocation and has found the specified name refers to a JCL DD card, not a template. In all cases the presence of a DD card takes precedent over any TEMPLATE of the same name.

Severity: 0 (informational)

System action: PREVIEW processing continues.

User response: If a TEMPLATE and DD card have the same name, the DD card always takes precedence. If the indicated name should not reference a DD card, perform one of the following actions:

- Remove the DD card
- Change the TEMPLATE name on the TEMPLATE control statement and the utility control statement that refers to it

If the reference to the DD card is correct, no action is necessary. Refer to *DB2 Utility Guide and Reference* for the correct option specifications.

DSNU1026I *csect-name* **- LIST** *using-list* **REUSES LIST** *used-list*

Explanation: A LISTDEF list is INCLUDING or EXCLUDING another list that has already been used in expansion of this clause. This "circular reference" is not allowed. For example, the following specification is not allowed: `LISTDEF A INCLUDE LIST B  LISTDEF B INCLUDE LIST A`

Severity: 8 (error)

System action: Utility processing does not initiate.

User response: Correct one or more LISTDEF control statements to eliminate the circular reference and resubmit the utility job. Refer to *DB2 Utility Guide and Reference* for the correct option specifications.

DSNU1027I *csect-name* **- PROCESSING CONTINUES DUE TO OPTIONS ITEMERROR SKIP**

Explanation: An error has occurred during utility processing, indicated by previous error messages. However, because an OPTIONS ITEMERROR SKIP control statement is in force, processing continues.

Severity: 8 (error) or 4 (warning)

System action: Utility processing continues on the
current utility control statement only. Subsequent utility
control statements in the same job step will not be
processed, unless reason code 4 is issued.

User response: Correct the previous error, and resubmit the utility for the objects in error. If the utility experiencing an error is QUIESCE and a single quiesce point is required for the entire list, the entire list must be resubmitted.

DSNU1028I *csect-name* **- INVALID KEYWORD** *kkkkkkkk* **FOR DASD TEMPLATE**

Explanation: A DASD TEMPLATE control statment cannot include the tape TEMPLATE keyword *kkkkkkkk*.

Severity: 8 (error)

System action: Utility processing does not initiate.

User response: DB2 uses the UNIT operand to determine the data set type. Either remove the tape keyword from the TEMPLATE control statement, or

correct the UNIT operand to refer to a tape unit. Refer to *DB2 Utility Guide and Reference* for the correct option specifications.

DSNU1029I *csect-name* **- INVALID KEYWORD** *kkkkkkkk* **FOR TAPE TEMPLATE**

Explanation: A tape TEMPLATE control statement cannot include the DASD TEMPLATE keyword *kkkkkkkk*.

Severity: 8 (error)

System action: Utility processing does not initiate.

User response: DB2 uses the UNIT operand to determine the data set type. Either remove the DASD keyword from the TEMPLATE control statement, or correct the UNIT operand to refer to a DASD unit. Refer to *DB2 Utility Guide and Reference* for the correct option specifications.

DSNU1030I *csect-name* **- QUALIFIER** *qqqqqqqq* **TOO LONG**

Explanation: The TEMPLATE DSN qualifier '*qqqqqqqq*' is greater than eight characters.

Severity: 8 (error)

System action: Utility processing for the current object halts. If a list is being processed, utility processing of the next object will proceed or halt based on the OPTIONS ITEMERROR control statement. By default execution will HALT.

User response: Correct the TEMPLATE DSN expression and resubmit the utility job. Refer to *DB2 Utility Guide and Reference* for the correct option specifications.

DSNU1031I *csect-name* **- DSN** *name* **TOO LONG**

Explanation: The TEMPLATE DSN operand results in a data set name '*name*' that is too long for the type of data set being allocated.

Severity: 8 (error)

System action: Utility processing for the current object halts. If a list is being processed, utility processing of the next object will proceed or halt based on the OPTIONS ITEMERROR control statement. By default execution will HALT.

User response: Correct the TEMPLATE DSN operand in error, and resubmit the utility job. Refer to *DB2 Utility Guide and Reference* for the correct option specifications.

DSNU1032I *csect-name* **- INVALID QUALIFIER** *qualifier* **IN TEMPLATE** *name*

Explanation: The TEMPLATE DSN qualifier either contains invalid characters or does not start with an alpha character.

Severity: 8 (error)

System action: Utility processing for the current object halts. If a list is being processed, utility processing of the next object will proceed or halt based on the OPTIONS ITEMERROR control statement. By default, execution will HALT.

User response: Correct the TEMPLATE DSN operand in error, and resubmit the utility job. A common cause of this error message is leaving the trailing period off of a TEMPLATE DSN variable name. All TEMPLATE DSN variable names are of the form `&xyz.` and must contain the trailing period. Refer to *DB2 Utility Guide and Reference* for the correct option specifications.

DSNU1033I *csect-name* **- PROCESSING LIST ITEM:** *object-type qualifier.name*

Explanation: While processing a LISTDEF list, utility processing has begun on the indicated object.

Severity: 0 (informational)

System action: Utility processing continues.

DSNU1034I *csect-name* **-** *primary-secondary* **SPACE** *units quantity* **TOO LARGE FOR DATA SET** *name*

Explanation: The TEMPLATE PRIMARY or SECONDARY SPACE quantity indicated is too large. It is not possible to dynamically allocate a data set larger than 16 777 215 units.

Severity: 8 (error)

System action: Utility processing for the current object halts. If a list is being processed, utility processing of the next object will proceed or halt based on the OPTIONS ITEMERROR control statement. By default, execution will HALT.

User response: The DB2-calculated required space is too large to be dynamically allocated in the units shown. If the primary allocation is too large, use the PCTPRIME keyword to reduce the primary and increase the secondary. If the secondary allocation is too large, use the NBRSECND keyword to increase the number of secondary allocations. It is not possible to use TEMPLATE dynamic allocation for data sets larger than 16×(16 777 215) MB. If an object requires a data set of this size it must be processed individually using a DD card, not a template.

DSNU1035I *csect-name* **-** *statement-type* **STATEMENT PROCESSED SUCCESSFULLY**

Explanation: DB2 issues this message following the successful execution of an OPTIONS, LISTDEF, or TEMPLATE utility control statement.

Severity: 0 (informational)

System action: DB2 processed and retained the options or definitions specified on the control statement for use on subsequent utility control statements.

DSNU1036I *csect-name* **- UNABLE TO ESTIMATE SPACE REQUIREMENTS FOR** *dataset-use*

Explanation: While processing a TEMPLATE control statement which does not include a SPACE keyword, DB2 could not calculate the default data set size. DB2 bases default calculations on the size of existing data sets. In this situation, DB2 could not determine the size of existing data sets because one or more of those existing data sets are:

- in-stream data sets,
- members of partitioned data sets,
- on tape,
- or results sets generated by DECLARE CURSOR executed with an EXEC SQL utility statement.

Severity: 8 (error)

System action: DB2 halts utility processing for the current object. If the utility is processing a list of objects, utility processing of the next object will proceed or halt based on the OPTIONS ITEMERROR control statement. By default, processing will HALT.

User response: Add the SPACE keyword and its operands to the TEMPLATE control statement in error and rerun the utility. If the utility is the LOAD utility, you must add SPACE keywords for all TEMPLATE control statements referenced in the LOAD, except for the input data set.

DSNU1037I *csect-name* **- PREVIEW PROCESSING CONTINUES DUE TO JCL PARAMETER**

Explanation: An OPTIONS utility control statement has attempted to turn off PREVIEW processing. PREVIEW processing continues because the PREVIEW JCL parameter has been specified. This parameter applies to the entire job step and can not be overridden by an OPTIONS control statement.

Severity: 0 (informational)

System action: Utility processing continues in PREVIEW mode.

User response: If PREVIEW processing must be turned off in the middle of a utility job step, remove the PREVIEW JCL parameter. Activate or deactivate PREVIEW processing by using the OPTIONS utility control statement, not the PREVIEW JCL parameter.

DSNU1038I *csect-name* **- DATASET ALLOCATED. TEMPLATE=***template-name* **DDNAME=***dd-name* **FILE SEQUENCE=***file-sequence* **DSN=***ds-name*

Explanation: The dataset that was indicated in the message has been successfully allocated based on a TEMPLATE control statement. *file-sequence* will only be shown for tape datasets.

Severity: 0 (informational)

System action: Processing continues.

User response: Refer to Part 4 (Volume 1) of the *DB2 Administration Guide* for more information.

DSNU1039I *csect-name* **- PROCESSING LIST ITEM:** *object-type qualifier.name* **PARTITION** *partnum*

Explanation: While processing a LISTDEF list, utility processing has begun on the indicated object.

Severity: 0 (informational)

System action: Utility processing continues.

DSNU1040I *csect-name* **- UNABLE TO LOCATE DSN** *dsname* **ON RESTART**

Explanation: DB2 could not find data set *dsname* when the utility was restarted. The data set is required for restart to continue.

System action: The utility step abends. The utility job step is in the stopped state.

User response: Refer to the DB2 abend code x'00E400C3' for more information.

DSNU1041I *csect-name* **- UNABLE TO CREATE GDG BASE** *name reason*

Explanation: A TEMPLATE could not allocate a generation data group (GDG) data set, but the base GDG did not exist. The utility was unable to define the GDG base.

Severity: 8 (error)

System action: Utility processing halts for the current object. If a list is in process, utility processing of the next object will proceed or halt based on the OPTIONS ITEMERROR control statement. By default, execution will HALT.

User response: Review the reason value to determine why the GDG base could not be defined. The possible reasons and suggested responses include:

- UNABLE TO LOAD IDCAMS PROGRAM. Ensure that the utility can invoke the IDCAMS program. Ensure that the utility can invoke the IDCAMS program.
- IDCAMS ABEND *xxx*. Use *xxx*, the abend code from the IDCAMS program, to determine why the IDCAMS program abended.
- ERROR IN IDCAMS. Review the IDCAMS message output that is listed between messages DSNU1042I and DSNU1043I to determine why the GDG base could not be defined.

DSNU1042I *csect-name* **- START OF IDCAMS MESSAGES**

Explanation: This message indicates the beginning of the IDCAMS messages or other related non-DB2 messages. DSNU1043I indicates the end.

User response: No action is required, but subsequent IDCAMS messages may require some action. See the *z/OS MVS System Messages Volumes 1-10* or other appropriate documentation for more information.

DSNU1043I *csect-name* **- END OF IDCAMS MESSAGES**

Explanation: This message indicates the end of the IDCAMS or other related non-DB2 messages. DSNU1042I indicates the beginning.

User response: No action is required, but preceding messages may require some action. See the *z/OS MVS System Messages Volumes 1-10* for more information.

DSNU1044I *csect-name* **- PROCESSING** *ddname* **AS EBCDIC**

Explanation: The utility control statement input data set begins with an EBCDIC character and is assumed for the duration of this job step to contain only EBCDIC characters.

Severity: 0 (informational)

System action: Processing of EBCDIC utility control statements continues.

User response: If processing of utility control statements as EBCDIC is incorrect make sure the first character in the input data set is UNICODE, specifically UTF-8 CCSID 1208. This will cause the entire input data set to be processed as UNICODE.

DSNU1045I *csect-name* **- PROCESSING** *ddname* **AS UNICODE**

Explanation: The utility control statement input data set begins with a UNICODE character, specifically UTF-8 CCSID 1208, and is assumed for the duration of this job step to contain only UNICODE characters.

Severity: 0 (informational)

System action: Processing of UNICODE utility control statements continues.

Attention: UTF-8 quoted strings may not print correctly in utility error messages. All messages are printed in EBCDIC, and the quoted strings remain in UTF-8, causing the output to display poorly in EBCDIC.

User response: If processing of utility control statements as UNICODE is incorrect make sure the first character in the input data set is EBCDIC. This will cause the entire input data set to be processed as EBCDIC.

DSNU1050I *csect-name* **- INVALID SUBSTRING NOTATION IN EXPRESSION** *string* **ON TEMPLATE** *template name*

Explanation: Invalid substring notation was detected on the TEMPLATE DSN operand. The portion of the DSN operand which is in error and the TEMPLATE name on which it occurred are shown.

Severity: 8 (error)

System action: Utility processing is terminated.

User response: Correct the TEMPLATE control statement which is in error. Common causes of this error are incorrect syntax of the (s,l) clause and values of ″s″ or ″l″ which fall outside of the variable value at the time of execution.

DSNU1051I *csect-name* **TEMPLATE SUBSYS IS NOT SUPPORTED TO ALLOCATE FILE FOR** *util-name* **UTILITY**

Explanation: Dynamically allocating a BatchPipes file is not supported for *util-name* UTILITY.

Severity: 8 (error)

User response: Remove TEMPLATE keyword, SUBSYS, and resubmit the job.

DSNU1052I *csect-name util-name* **UTILITY DOES NOT SUPPORT RESTART WHEN TEMPLATE SUBSYS IS USED**

Explanation: The utility that uses the BatchPipes file cannot be restarted.

Severity: 8 (error)

User response: Use the DB2 command TERM UTILITY to terminate the stopped utility, and then rerun the job.

DSNU1053I *csect-name* **KEYWORD** *k1* **INVALID WITH KEYWORD** *k2* **ON** *utility-name*

Explanation: The use of KEYWORD *k1* is not allowed due to a conflicting keyword *k2* on another utility control statement. The conflicting LISTDEF or TEMPLATE name is shown.

Severity: 8 (error)

User response: Correct one or both control statements
in error and resubmit the job.

DSNU1100I *csect-name* **RECORD 'n' FOR PARTITION** *m* **WILL BE DISCARDED DUE TO** *field-name* **CONVERSION ERROR FOR** *table-name*

Explanation: The load utility found the input data provided for a field to be incorrect. The corresponding input data set record 'n' from the input data set specified for partition *m* will be placed in the DISCARDS data set.

Severity: 4 (warning)

System action: The input data set record is noted and will be discarded in the DISCARD phase. The LOAD utility continues to process additional records in the input data set.

User response: Following the completion of the current LOAD utility invocation, examine the record in the DISCARDS data set for partition *m*. Correct the error in the discarded input data. Submit a RESUME(YES) LOAD invocation with the DISCARDS data set as the input data set. Refer to *DB2 Utility Guide and Reference* for a definition of LOAD input field column specifications.

DSNU1101I *csect-name* **RECORD 'n' FOR PARTITION** *m* **WILL BE DISCARDED DUE TO INVALID** *field-name* **COLUMN SPECIFICATION**

Explanation: The load utility calculated the location of an input data value using the field specifications supplied and VARCHAR length fields present within the input data set record. The calculated position for the field was determined to be incorrect.

Severity: 4 (warning)

System action: The input data set record number is noted and will be discarded in the DISCARD phase. The LOAD utility continues to process additional records in the input data set.

User response: Following the completion of the current LOAD utility invocation, examine the record in the DISCARDS data set for partition *m*. Correct the error in the discarded input data, if one exists. Otherwise, correct the input field specification. Submit a RESUME(YES) LOAD invocation, which uses the DISCARDS data set and the verified LOAD utility statement and field specifications. Refer to *DB2 Utility Guide and Reference* for a definition of LOAD input field column specifications.

DSNU1102I *csect-name* **RECORD 'n' FOR PARTITION** *m* **WILL BE DISCARDED DUE TO ERROR = X'reason-code' FROM EXIT ROUTINE** *procedure-name* **FOR** *table-name*

Explanation: The exit routine provided by your installation blocked the load of record 'n' for partition *m* into the table indicated. The reason-code was set by the exit routine.

Severity: 4 (warning)

System action: The input data set record number is noted and will be discarded in the DISCARD phase. The record in error is placed in the DISCARDS data set. The LOAD utility continues to process additional records in the input data set.

User response: Following the completion of the current LOAD utility invocation, examine the record in the DISCARDS data set for partition *m*. Correct the error in the discarded input data, if one exists. Ensure that the exit routine provided is correct. Submit a RESUME(YES) LOAD invocation, which uses the DISCARDS data set and the verified LOAD utility statement and field specifications. Refer to *DB2 Utility Guide and Reference* for a definition of LOAD input field column specifications.

System programmer response: Verify that the validation routine is correct.

DSNU1103I *csect-name* **RECORD 'n' FOR PARTITION** *m* **WILL BE DISCARDED DUE TO PART OR HIGH KEY VALUESPECIFICATION(S)**

Explanation: The discarded record does not belong to the partition *m* being loaded. For a LARGE table space, this might be because the key is higher than the high key value specified for the last partition.

Severity: 4 (warning)

System action: The input data set record number is noted and will be discarded in the DISCARD phase. The LOAD utility continues to process additional records in the input data set.

User response: Following the completion of the current LOAD utility invocation, examine the record in the DISCARDS data set for partition *m*. Correct the error in the discarded input data, if one exists. Submit a RESUME(YES) LOAD invocation with the DISCARDS data set as the input data set. Refer to Part 2 of *DB2 Utility Guide and Reference* for a definition of LOAD input field column specifications.

System programmer response: Verify that the validation routine is correct.

DSNU1104I *csect-name* **RECORD 'n' FOR PARTITION** *m* **WILL BE DISCARDED DUE TO WHEN SPECIFICATION(S)**

Explanation: The discarded record does not satisfy the WHEN selection expression provided for partition *m*.

Severity: 4 (warning)

System action: The input data set record is noted and will be discarded in the DISCARD phase. The LOAD utility continues to process additional records in the input data set.

User response: Following the completion of the current LOAD utility invocation, examine the record in the DISCARDS data set for partition *m*. Correct the error in the discarded input data, if one exists. Submit a RESUME(YES) LOAD invocation with the DISCARDS data set as the input data set. Refer to *DB2 Utility Guide and Reference* for a definition of LOAD input field column specifications. If FORMAT UNLOAD was specified on the load statement, the WHEN specifications are built internally. This ensures that the rows are loaded back into the same object from which they were unloaded. Refer to *DB2 Utility Guide and Reference* for a description of the FORMAT UNLOAD option of LOAD.

DSNU1105I *csect-name* **RECORD 'n' FOR PARTITION** *m* **WILL BE DISCARDED DUE TO UNEXPECTED ERROR**

Explanation: The LOAD of input data set record number 'n' for partition *m* was attempted but was not successful due to an unexpected error.

Severity: 8 (error)

System action: If the error was caused because a resource was not available, a DSNT500I message will be issued.

The utility step will be abended with a system X'04E' and an abend reason code of '00E40318'. If a DSNT500I message is not issued, an SVC dump will be requested. A SYSABEND dump of batch storage is also requested.

User response: If a DSNT500I message was issued, refer to the description under that message number. If the unavailable resource becomes available, the utility step is restartable.

If there is no DSNT500I message, the error is an unexpected internal error and you should consult your system programmer.

Operator response: An operator with SYSOPR authority can issue a TERM UTIL to free any held resources if that is necessary.

System programmer response: Refer to *DB2 Command Reference* for information about the TERM UTIL command. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNU1106I *csect-name* **RECORD 'n' FOR PARTITION** *m* **WILL BE DISCARDED DUE TO INVALID** *field-name* **COLUMN SPECIFICATION FOR** *table-name*

Explanation: The load utility calculated the location of an input data value using the field specifications supplied and VARCHAR length fields present within the input data set record 'n' for partition *m*. The calculated position for the field was determined to be incorrect.

Severity: 4 (warning)

System action: The input data set record number is noted and will be discarded in the DISCARD phase. The LOAD utility continues to process additional records in the input data set.

User response: Following the completion of the current LOAD utility invocation, examine the record in the DISCARDS data set for partition *m*. Correct the error in the discarded input data, if one exists. Otherwise, correct the input field specification. Submit a RESUME(YES) LOAD invocation, which uses the DISCARDS data set and the verified LOAD utility statement and field specifications. Refer to *DB2 Utility Guide and Reference* for a definition of LOAD input field column specifications.

DSNU1107I *csect-name* **- UNIQUE INDEX KEY DUPLICATES KEY OF INDEXED ROW AT RID** *rrrr* **INDEX =** *index-name* **TABLE =** *table-name* **RECNO =** *input-dataset-record-number* **FOR PART =** *m* **RID =** *duprid*

Explanation: The input data set record indicated was loaded into table 'table-name', partition *m*, at RID 'duprid', but it causes a unique index violation for index 'index-name'. It duplicates the key value of a record already in the table at RID 'rrrr'.

Severity: 4 (warning)

System action: Processing continues. No index entry is created for the record at RID 'duprid'. The record will be deleted in a subsequent phase of processing.

DSNU1108I *csect-name* **UNIQUE INDEX KEY DUPLICATES KEY FROM INPUT DATASET RECORD** *iiii* **FOR PART** *m* **LOADED AT RID** *rrrr* **INDEX =** *index-name* **TABLE =** *table-name* **RECNO =** *recno* **RID =** *duprid*

Explanation: Input data set record 'recno' was loaded into table 'table-name', partition *m*, at RID 'duprid', but it causes a unique index violation for index 'index-name'. It duplicates the key value of another record in the load job: the record loaded at RID 'rrrr' from input data set record 'iiii'.

Severity: 4 (warning)

System action: Processing continues. Index entries are not created for either record. Both records will be deleted in a subsequent phase of processing.

DSNU1109I *csect-name* **RECORD '***n***' FOR PARTITION** *m* **WILL BE DISCARDED DUE TO CHECK CONSTRAINT** *constraint-name* **VIOLATION ON TABLE** *table-name*

Explanation: The discarded record violates the check constraint named *constraint-name* defined on table *table-name*. The record is not loaded and is discarded during the DISCARD phase.

Severity: 4 (warning)

System action: The LOAD utility continues to process additional records in the input data set.

User response: After the LOAD utility completes, examine the record in the DISCARDS data set for partition *m*. Correct the error in the discarded input data. Submit a RESUME(YES) LOAD job with the DISCARDS data set as the input data set.

DSNU1110I *csect-name* **RECORD '***n***' FOR PARTITION** *m* **VIOLATES CHECK CONSTRAINT** *constraint-name* **ON TABLE** *table-name*

Explanation: The LOAD utility cannot load record *n* into table *table-name* partition *m* because the check constraint *constraint-name* would be violated. The record is not loaded.

Severity: 8 (error)

System action: Processing terminates. An abend is requested so that the LOAD can be restarted after the error condition is corrected.

User response: Find the failing record in the input data set for partition *m*, correct the condition that caused the check constraint to be violated, and restart the LOAD utility.

DSNU1111I *csect-name* **FOREIGN KEY HAS NO PRIMARY KEY FOR RELATIONSHIP=***r1* **INPUT RECORD NUMBER=***nnnn***, PART** *m***, RID=X'rrrr'**

Explanation: Input record 'nnnn' for partition *m* caused a row to be loaded into table 'table' at RID 'rrrr' with no primary key for the relationship with the name 'r1'.

Severity: 4 (warning)

System action: Processing continues.

User response: None required. The constraint violation described will be rectified later in processing.

DSNU1112I *csect-name* **DISCARD PHASE STATISTICS -** *nnn* **INPUT DATA SET RECORDS DISCARDED FOR PARTITION** *m*

Explanation: Input data set records were rejected by the LOAD utility for one or several reasons. The rejected records have been placed in a discards data set. The number of records rejected for partition *m* is 'nnn'. Message DSNU399I summarizes the errors causing the records to be rejected.

Severity: 4 (warning)

System action: Processing continues.

User response: Examine the LOAD error summary and decide which of the discarded records, if any, should be corrected and reapplied to the table space.

DSNU1113I *csect-name* **- LOAD UTILITY ERROR SUMMARY REPORT**

Explanation: This message summarizes errors detected during the load job. A report is issued only if unique index or referential constraint violations were detected. The report contains one line for each error. The errors are listed first by error severity and within that by *idsrn*.

The report fields have the following meanings:

sev

The severity of the error reported on this line:

Severity	Type of error	Explanation
1	Primary	Primary referential integrity errors and all nonreferential integrity errors.
2	Secondary	Secondary referential integrity errors (that is, errors that result from a primary referential integrity error in an ancestor).

idsrn The relative record number in the input data set, for the partition identified by *pno*, of the record causing the error reported on this line.

pno The partition number for the input data set of the record causing the error reported on this line.

ddsrn The relative record number in the input data set of the record causing the error reported on this line, if discard processing is in effect. Otherwise, *ddsrn* is blank.

type A description of the type of error reported on this line, as follows:
Conversion
Field specification
Validation

Part Specification
Unexpected
Column specification
Referential integrity
Check constraint
Duplicate key.

name1 The name of the table for which the record causing the error was targeted, if this information is pertinent for the type of error reported on this line. Otherwise, *name1* is blank. This field has nonblank values for the following error types:
Conversion
Validation
Column specification
Referential integrity (primary).
Check constraint

name2 The name of the field or fanset involved in the error reported on this line, if this information is pertinent for the type of error reported. Otherwise, *name2* is blank. This field contains a field name for the following types of errors:
Conversion
Field specification
Column specification.

For primary referential integrity errors, *name2* is a relationship name. For check constraint errors, *name2* is a check constraint name.

idsrn2 The input data set record number of a related error. For duplicate key errors, it gives the input data set record number of the record whose key is duplicated. For secondary referential integrity errors, it gives the input data set record number of the record whose primary referential integrity error cascaded to cause this secondary error. For other types of errors, this field is unused.

Severity: 4 (warning)

System action: Processing continues.

DSNU1114I *csect-name* **(RE)LOAD PHASE STATISTICS NUMBER OF RECORDS LOADED =** *nnnn* **FOR TABLE** *table-name*

Explanation: When the SHRLEVEL CHANGE option is selected, DB2 issues this message upon normal completion of the (RE)LOAD phase of the LOAD utility. It indicates the number of records loaded into TABLE *table-name* from the scource record data set (that is, the data set identified by the JCL DD statement identified in the INDDN option for the LOAD. This message is repeated for each TABLE loaded by the current LOAD utility statement

Severity: 0 (informational)

System action: Processing of the RELOAD phase continues.

User response: Verify that the number of record loaded reported is consistent with expectations. For example, if four records are processed, the user should expect to see NUMBER OF RECORD LOADED = 4.

DSNU1115I *csect-name nnnn* **RESTART PHASE NOT ALLOWED IN THE (RE)LOAD PHASE, PHASE CHANGED TO CURRENT**

Explanation: This message is issued when the RESTART PHASE is requested for a LOAD SHRLEVEL CHANGE utility. RESTART PHASE is not allowed for LOAD SHRLEVEL CHANGE.

Severity: 4 (warning)

System action: Processing of the RELOAD phase continues.

User response: No action is necessary. If another RESTART is required, use the RESTART option instead of RESTART PHASE.

DSNU1116I *csect-name* **FOREIGN KEY HAS NO PRIMARY KEY FOR RELATIONSHIP=***r1* **INPUT RECORD NUMBER=***nnnn*

Explanation: Input record *nnnn* has no primary key for the relationship with the name *r1*. The row is not loaded.

Severity: 4 (warning)

System action: Processing continues.

User response: None required.

DSNU1117I *csect-name* **- UNIQUE INDEX KEY DUPLICATES KEY OF INDEXED RECORD AT RID** *rrrr* **INDEX =** *index-name* **TABLE =** *table-name* **RECNO =** *input-dataset-record-number*

Explanation: The input data set record was loaded into table *table-name* at RID *duprid*, but it causes a unique index violation for index *index-name*. It duplicates the key value of a record already in the table at RID *rrrr*.

Severity: 4 (warning)

System action: Processing continues. No index entry is created for the record at RID *duprid*.

DSNU1118I *csect-name* **(RE)LOAD PHASE STATISTICS -** *nnnn* **DUPLICATE KEY ERRORS ENCOUNTERED**

Explanation: This message is issued by the (RE)LOAD phase of the LOAD utility when the SHRLEVEL CHANGE option is used. The number of duplicate key errors that were encountered during processing is *nnnn*.

Severity: 0 (informational)

System action: Processing continues.

DSNU1120I AUX TABLE NOT DEFINED FOR LOB TABLE SPACE *dbname.tsname*

Explanation: The LOB table space *dbname.tsname* has been created, but no auxiliary table has been created in it.

Severity: 8 (error)

System action: Utility operation is terminated with return code 8.

User response: Complete the definition of the LOB table space before resubmitting the utility.

DSNU1121I *csect-name* **(RE)LOAD PHASE STATISTICS - NUMBER OF RECORDS LOADED =** *nnnn* **FOR TABLE** *table-name* **PART** *m*

Explanation: When the SHRLEVEL CHANGE option is selected, DB2 issues this message upon normal completion of the (RE)LOAD phase of the LOAD utility. It indicates the number of records that were loaded into part *m* of table *table-name* from the source record data set. The message repeats for each INTO TABLE PART clause in the LOAD utility statement.

Severity: 0 (informational)

System action: Processing continues.

User response: Verify that the number of records loaded reported is consistent with expectations.

DSNU1122I JOB *job-name* **PERFORMING** *util-type* **WITH UTILID** *util-id* **UNABLE TO DRAIN** *object-name***. RETRY** *n* **OF** *m* **WILL BE ATTEMPTED IN** *s* **SECONDS**

Explanation: A utility with the RETRY option specified has been unable to drain a tablespace, tablespace part, or index. The utility will make another attempt to acquire the drain after a delay.

Severity: 0 (informational)

System action: The utility job continues.

User response: The utility job attempts to acquire a restrictive access to the referenced object, and will retry if unsuccessful. Optionally, the user may determine who has claims on the object via the -DIS DATABASE command with CLAIMERS option, and cancel or end those threads.

DSNU1123I LOGICAL PARTITIONS WILL BE LOADED IN PARALLEL, NUMBER OF TASKS = nnnn

Explanation: This message is issued by the BUILD2 phase of the REORG utility. The message indicates that the REORG utility is able to update logical partitions in parallel. The number of utility tasks used to update logical partitions is indicated by :mv.nnnn:emv.. .*:p.This message is issued by the following CSECT(s): .*DSNURBD2

Severity: 0

System action: Processing of the BUILD2 phase continues.

Programmer response: You can verify that the number of tasks reported is consistent with expectations. For example, if 4 logical partitions are to be updated, the user might expect to see number of tasks = 4.

DSNU1124I *csect-name* **- UNIQUE INDEX KEY DUPLICATES KEY OF INDEXED RECORD AT RID** *rrrr* **INDEX =** *index-name* **TABLE =** *table-name* **RECNO =** *input-dataset-record-number* **FOR PART =** *m*

Explanation: The input data set record indicated for PART *m* of table *table-name* causes a unique index violation for index *index-name*. It duplicates the key value of a record already in the table at RID *rrrr*. The record was not loaded.

Severity: 4 (warning)

System action: Processing continues.

DSNU1125I *csect-name* **FOREIGN KEY HAS NO PRIMARY KEY FOR RELATIONSHIP=***r1* **INPUT RECORD NUMBER=***nnnn***,FOR PART** *m*

Explanation: Input record *nnnn* for partition *m* for table *table* has no primary key for the relationship with the name *r1*. The row is not loaded.

Severity: 4 (warning)

System action: Processing continues.

User response: None required.

DSNU1126I *csect-name* **PARTITION** *n* **IS SPECIFIED IN MORE THAN ONE INTO TABLE PART CLAUSE, PARTITION PARALLELISM IS DISALLOWED**

Explanation: LOAD partition parallelism was requested via INDDN or INCURSOR following INTO TABLE PART clauses. However, the same partition was specified on more than one INTO TABLE PART clause. Partition parallelism is not supported for this case.

Severity: 8 (error)

System action: Utility operation is terminated with return code 8.

User response: Correct the LOAD statement in one of the following ways:

- Specify a different partition number on each INTO TABLE PART clause
- Remove the parallelism by removing the INDDN or INCURSOR keywords from the INTO TABLE PART clauses.

DSNU1127I *csect-name op* **OPERATION FOUND ON RECORD REMOVED BY DISCARD CRITERIA**

Explanation: Operation *op* was detected during a REORG SHRLEVEL CHANGE on a record removed during the UNLOAD phase.

Severity: 8 (error)

System action: Utility processing is terminated.

User response: Rerun the utility when data being discarded by REORG is not being updated or deleted by applications or SQL.

DSNU1128I *csect-name* **FEWER PAGES THAN PARTS FOR REBALANCE**

Explanation: In attempting to rebalance partitions during REORG, it is estimated that there are fewer total pages than the number of partitions being rebalanced.

Severity: 8 (error)

System action: Remove the REBALANCE keyword form the REORG until there is sufficient data to populate pages for every partition.

DSNU1129I *csect-name* **- PARTITION RANGE NOT CONTIGUOUS - REBALANCE IGNORED**

Explanation: The partition range that was specified on a REORG REBALANCE does not represent a set of contiguous logical partitions. Due to a series of ALTER TABLE ADD PART and ALTER TABLE ROTATE PART statements, the physical partition numbers do not match the logical partition numbers.

The physical partition numbers that were specified as the REORG partition range do not represent a contiguous set of logical partitions. For example, physical partitions 1 and 2 might represent logical partitions 2 and 4.

Severity: 8 (error)

System action: The REORG terminates with return code 8.

User response: Use the DISPLAY DATABASE command to identify a range of physical partition numbers that represent a contiguous set of logical partitions, and resubmit the REORG REBALANCE while specifying that physical partition range. Alternatively, omit the partition range on the REORG REBALANCE so that the entire table space is reorganized.

DSNU1130I *csect-name* **- NOT ALL PARTITIONS POPULATED BY REBALANCE - PROCESSING TERMINATES**

Explanation: The table space that is being reorganized has many rows with duplicate partitioning key values, and there were insufficient unique values to allow distributing rows across all partitions during a rebalance.

Severity: 8 (error)

System action: The utility is terminated with return code 8. For REORG SHRLEVEL NONE, the table space is left in the recover pending restricted state.

User response: If the REORG was REORG SHRLEVEL NONE, recover the table space to the point in time before the REORG.

DSNU1131I *csect-name* **NOT ENOUGH DATA FOR REMAINING COLUMNS STARTING WITH COLUMN** *colno* **IN** *creator.tablename*

Explanation: The REORG utility has discovered that there is not enough data to fill out the remaining column or columns in a row.

Severity: 8 (error)

System action: The REORG utility is abended with RC00E40349. An SDUMP might be taken.

User response: Notify the system programmer.

Problem determination: The probable cause is that the table was not originally loaded correctly; there was not enough data to fill out all of the columns. This error can occur when DSN1COPY is used to copy one table to another table that was not defined correctly. To correct this error:

1. Define a new table with the same column definitions as the original table.
2. Use ALTER TABLE ADD COLUMN to add any additional columns to the new table.
3. Run the REORG utility on the table space containing the new table so the data matches the column definitions.
4. Run DSN1COPY to populate the new table.

If this does not correct the error, collect the following diagnostic items listed in Appendix A, “Problem determination,” on page 659: 1, 5, 32, 46, 60, 61

DSNU1141I *csect-name* **INVALID DELIMITER CHARACTER** *x*

Explanation: The delimiter character *x*, which was specified as the operand of a COLDEL, CHARDEL, or DECPT keyword, is not a valid delimiter.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Review the rules for valid delimiters as stated in the *DB2 Utility Guide and Reference*. Correct the keyword and resubmit the job.

DSNU1142I *csect-name* **RECORD** *n* **DELIMITER MISMATCH FOR COLUMN** *colname*

Explanation: A column or character delimiter character that marks the end of input for column *colname* was not found before the end of the record.

Severity: 8 (error)

System action: Processing terminates. An abend is requested so that the LOAD can be restarted after the error is corrected.

User response: Find the failing record in the input data set, correct the condition that caused the error, and restart the LOAD utility.

DSNU1143I *csect-name* **RECORD** *n* **WILL BE DISCARDED DUE TO DELIMITER MISMATCH FOR COLUMN** *colname*

Explanation: A column or character delimiter character that marks the end of input for column *colname* was not found before the end of the record. The record is not loaded, and is discarded during the DISCARD phase.

Severity: 4 (warning)

System action: The LOAD utility continues to process records in the input data set.

User response: After the LOAD utility completes, examine the records in the discard data set. Correct the errors in the discarded input data, and submit a LOAD RESUME YES job with the DISCARD data set as the input.

DSNU1145I *csect-name* **FORMAT DELIMITED REQUIRES ONE INTO TABLE**

Explanation: LOAD FORMAT DELIMITED supports loading only a single table. LOAD FORMAT DELIMITED was specified in a LOAD statement with more than one INTO TABLE clause.

Severity: 8 (error)

System action: Processing terminates.

User response: Correct the LOAD statement and resubmit the utility.

DSNU1146I *csect-name* **A MAXIMUM OF** *max-parts* **ARE ALLOWED TO BE PROCESSED WHEN A TABLE SPACE IS COMPRESSED UNLESS THE PART KEYWORD IS USED.**

Explanation: A default limit of *max-parts* parts is used for the LOAD and REORG utilities if the object is compressed. The limit does not apply if a PART range is specified in the REORG control statements or if the PART keyword is specified on every INTO TABLE statement for LOAD. *max-parts* can be set using the MAX_UTIL_PARTS installation parameter.

Severity: 8 (error)

System action: Utility processing is terminated.

User response: Change the utility control statement to specify a PART range for REORG or use a PART keyword on every INTO TABLE statement for LOAD.

DSNU1147I *csect* **- (RE)LOAD PHASE STATISTICS - TOTAL NUMBER OF RECORDS LOADED=***nnnn* **FOR TABLESPACE** *dbname.tsname*

Explanation: This message is issued at the normal completion of the RELOAD phase of the LOAD utility. It indicates the total number of records loaded into the table space.

Severity: 0

System action: The utility continues.

User response: The user can verify that the number of records matches expectations. The total includes all records loaded into all tables for a segmented or linear table space, and all records loaded into partitions for a partitioned table space.

DSNU1148I *csect* **- (RE)LOAD PHASE STATISTICS - NUMBER OF INPUT RECORDS PROCESSED=***nnnn* **FOR PART** *n*

Explanation: The message is issued from the LOAD utility when LOAD partition parallelism is being used and the loading of partition *n* has been completed. The message indicates the total number of records that were processed from the input file for the partition. The message is issued for each partition being loaded.

Severity: 0

System action: The utility continues.

User response: The user can verify that the number of records matches expectations.

DSNU1149I *csect* **- (RE)LOAD PHASE STATISTICS - NUMBER OF INPUT RECORDS NOT LOADED=***nnnn* **FOR PART** *n*

Explanation: The message is issued from the LOAD utility when LOAD partition parallelism is being used and the loading of partition *n* has been completed. It indicates the total number of records from the input file for partition *n* that were not loaded. The records might not have been loaded because they were excluded by a WHEN clause, because their partitioning key did not

match partition *n*, or because there was an error processing the record.

The message is not issued if *nnnn* is zero, or if the job was restarted from the last commit point.

Severity: 0

System action: The utility continues.

User response: The user can verify that the number of records matches expectations. For example, if this message appears it might be an indication that some rows in the input file belong in other partitions, and should have been loaded into those partitions.

DSNU1150I *csect* **- (RE)LOAD PHASE STATISTICS - NUMBER OF INPUT RECORDS NOT LOADED=***nnnn*

Explanation: The message is issued at the normal completion of the RELOAD phase of the LOAD utility. It indicates the total number of records from the input file or files that were not loaded. The records might not have been loaded for one of the following reasons:

- They were excluded by a WHEN clause
- They were being loaded into a specific partition with INTO TABLE PART but their partitioning key did not match that partition
- There was an error processing the record

The message is not issued if *nnnn* is zero, or if the job was restarted from the last commit point.

Severity: 0

System action: The utility continues.

User response: The user can verify that the number of records matches expectations. For example, when loading into a single partition, if this message appears it might be an indication that some rows in the input file belong in other partitions, and should have been loaded into those partitions.

DSNU1178I *csect-name* **- SUM OF INPUT COLUMN LENGTHS FOR CURSOR** *cursor-name* **IS TOO LARGE**

Explanation: One of two possible errors occurred:

- The size of the record that would be loaded for cursor *cursor-name* is greater than 32767 bytes. DB2 calculates the sum of column lengths by adding the size of the non-LOB columns with 8 bytes per LOB column for the specified curser, *cursor-name*. If the sum of lengths is more than 32767, you receive this message.
- The sum of the lengths of the LOB columns is greater than the available storage.

Severity: 8 (error)

System action: Processing terminates.

User response: Depending on which reason the error occurred, complete one of the following actions:

- Decrease the number of columns or the lengths of individual columns so the sum of the lengths of the non-LOB columns (including 2 bytes for each nullable column) plus 8 bytes per LOB column does not exceed 32767.
- Increase the region size for the utility job.

DSNU1179I *csect-name* **THE SQL STATEMENT**
EXCEEDS THE MAXIMUM ALLOWED
LENGTH

Explanation: The statement cannot be executed
because it exceeds the system limits length.

Severity: 8 (error)

System action: The statement cannot be executed.

User response: Break the statement into shorter, or
less complex, SQL statements.

DSNU1180I *csect-name* **SQLCODE=0, SUCCESSFUL**
EXECUTION

Explanation: The SQLCODE was zero (0), indicating
successful execution of the function. SQLWARN
indicators might note some warnings. SQLWARNx
fields are described in Appendix C of *DB2 SQL*
Reference.

Severity: 0 (informational)

System action: Utility processing continues.

DSNU1184I *csect-name* **- SQLCODE =***sqlcode***,**
explanation

Explanation: This message contains an SQL return
code and a brief explanation, with the text inserted
from the SQLERRM field. The SQL return code is
positive, indicating a warning condition. For more
information about this SQL return code, see *DB2 Codes*.

Severity: 4 (warning)

System action: Utility processing continues.

DSNU1188I *csect-name* **- SQLCODE = -***sqlcode***,**
explanation

Explanation: This message contains an SQL return
code and a brief explanation, with the text inserted
from the SQLERRM field. The SQL return code is
negative, indicating that an error has occurred. For
more information about this SQL return code, see *DB2*
Codes.

Severity: 8 (error)

System action: Further utility processing is
terminated.

User response: Determine the cause for the SQL error

by using information about that specific SQLCODE. Correct the error. Rerun the application program or SQL statement.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 5, 14, 17.

DSNU1189I *csect-name* **- CURSOR** *cursor-name* **ALREADY DECLARED**

Explanation: The indicated *cursor-name* was declared previously in the SYSIN dataset.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Eliminate the duplicate DECLARE CURSOR and resubmit the utility job. Refer to *DB2 Utility Guide and Reference* for the correct option specifications.

DSNU1190I *csect-name* **- CURSOR** *cursor-name* **NOT DECLARED**

Explanation: A utility control statement used a cursor that was not found in SYSIN.

Severity: 8 (error)

System action: Utility processing is not initiated.

User response: Provide the missing DECLARE CURSOR in SYSIN prior to its use and resubmit the utility job. Refer to *DB2 Utility Guide and Reference* for the correct option specifications.

DSNU11191I *csect-name* **SQLCAID =** *xxxx*, **SQLCA IDENTIFIER**

Explanation: The SQLCA ID field (SQLCAID) is not the expected value, C'SQLCA'.

Severity: 0 (informational)

System action: Utility processing continues.

User response: The SQLCA area is invalid.

Problem determination: This error is an internal error.

DSNU11192I *csect-name* **SQLCABC =** *xxxx*, **SQLCA BYTE COUNT**

Explanation: The size of the SQLCA is not the expected value (136 bytes).

Severity: 0 (informational)

System action: Utility processing continues.

User response: The SQLCA area is invalid.

Problem determination: This error is an internal error.

DSNU11193I *csect-name* **SQLCODE =** *xxxx*, **INVALID SQLCODE**

Explanation: The SQL return code is not a valid value.

Severity: 0 (informational)

System action: Utility processing continues.

User response: Either the SQLCA area is invalid or the maintenance was not coordinated correctly. As a result, the SQL return code was added to the DB2 subsystem but not to the formatting routine. If the maintenance was not coordinated correctly, the DSNTIAM CSECT was probably not replaced in the current STEPLIB or JOBLIB.

Problem determination: If DSNTIAM is current then this error is an internal error.

DSNU11194I *csect-name* **SQLERRM =** *xxxx*, **SQLCA ERROR MESSAGE INSERTION TEXT**

Explanation: This message results from an SQL return code that is not recognized by DSNTIAR. SQLERRM contains error message test.

Severity: 0 (informational)

System action: Utility processing continues.

User response: Either the SQLCA area is invalid or the maintenance was not coordinated correctly. As a result, the SQL return code was added to the DB2 subsystem but not to the formatting routine. If the maintenance was not coordinated correctly, the DSNTIAM CSECT was probably not replaced in the current STEPLIB or JOBLIB.

Problem determination: If DSNTIAM is current then this error is an internal error.

DSNU11195I *csect-name* **SQLERRP =** *xxxx*, **SQL PROCEDURE DETECTING ERROR**

Explanation: The name of the procedure detecting the error is provided in the SQLERRP.

Severity: 0 (informational)

System action: Utility processing continues.

User response: Correct the error if one is noted in other messages. The SQL procedure detecting the error might be of help in diagnosing an error or warning. Rerun the SQL statement.

Problem determination: If DSNTIAM is current then this error is an internal error.

DSNU1196I **SQLERRD =** *xx,xx,xx,xx,xx,xx* **SQL DIAGNOSTIC INFORMATION**

Explanation: One or more of the six diagnostic fields do not have the expected value. Check that all of the

diagnostic fields contain one of the following expected values:

- 00, 00, 00, 00, 00, 00 or
- 00, 00, 00, -1, 00, 00

The SQL diagnostic information for each of the six values is described in the following list:

1 DB2 unique internal relational data system code

2 DB2 internal data manager error code

3 One of the following conditions:

- Number of rows inserted, updated, or deleted following an INSERT, UPDATE, or DELETE statement. **Attention:** A '-1' in this field indicates that a mass delete of a table in a segmented table space was performed, wherein DB2 does not access nor count individual rows.
- For SQLCODE -911 or -913, a reason code that indicates whether a deadlock or timeout occurred. **Tip:** The most common reason codes are:
  - 00C90088 - deadlock
  - 00C9009E - timeout

4 Reserved

5 Position or column of a syntax error for PREPARE or EXECUTE IMMEDIATE

6 DB2 internal buffer manager error code

Severity: 0 (informational)

System action: Utility processing continues.

User response: Refer to Appendix D of *DB2 SQL Reference* for further information on the contents of the SQLERRD fields. The SQL diagnostic information might help in diagnosing an error or in indicating the results of successful execution. If this error or warning occurs, correct the error. Rerun the program or SQL statement.

DSNU1197I SQLWARN*n-n* SQL WARNINGS

Explanation: At least one of the eleven (11) warning values is not the expected value (a blank). A nonblank value in one of the 11 warning fields has one of the following meanings:

SQLWARN0
Any other warning code is set.

SQLWARN1
String truncation.

SQLWARN2
Null values eliminated.

SQLWARN3
Number of host variables provided did not equal degree of the table.

SQLWARN4
No WHERE clause on UPDATE or DELETE.

SQLWARN5
Not a DB2 SQL statement.

SQLWARN6
Date or timestamp value adjusted.

SQLWARN7
Conversion of decimal number from precision 31 to precision 15 caused truncation of fractional digits.

SQLWARN8
Untranslatable character replaced with substitute.

SQLWARN9
Arithmetic exceptions have been ignored during processing of COUNT DISTINCT.

SQLWARNA
At least one character field of the SQLCA is invalid due to a string translation error.

Severity: 0 (informational)

System action: Utility processing continues.

User response: The SQL warning information might be of help in diagnosing an error or in indicating the results of successful execution. If this warning should occur, correct the error and rerun the program or SQL statement.

DSNU1198I SQLSTATE = *sqlstate*, SQLSTATE RETURN CODE

Explanation: The SQLSTATE is a return code that indicates the outcome of the most recently executed SQL statement. The execution of every SQL statement sets SQLSTATE to a five-digit code in the range of 00000 to 65535. The SQLSTATE has no effect on the existing use of any other field in the SQLCA.

Severity: 0 (informational)

System action: Utility processing continues.

User response: For further information on the meaning of an SQLSTATE return code, see the *DB2 Codes*. If necessary, correct the SQL statement and return it.

DSNU1199I *csect-name text*

Explanation: Unexpected message was returned from DSNTIAR.

Severity: 0 (informational)

System action: Utility processing continues.

User response: This error is an internal error.

DSNU1201I *csect-name* **- PARTITIONS WILL BE UNLOADED IN PARALLEL, NUMBER OF TASKS =** *nnnn*

Explanation: The message indicates that the UNLOAD utility is able to unload partitions in parallel. The number of utility tasks used to unload partitions is indicated by *nnnn*.

Severity: 0 (informational)

System action: Processing of the UNLOAD phase will continue.

DSNU1202I *csect-name* **- SAMPLING LIMIT HAS BEEN REACHED FOR TABLE** *table-name*

Explanation: In the UNLOAD statement, the LIMIT option is specified for table *table-name* and the number of unloaded rows from the table has reached the specified LIMIT value.

Severity: 0 (informational)

System action: No more rows will be unloaded from the table, but the unloading operation will continue for the previously unloaded tables if they exist.

DSNU1203I *csect-name* **- GENERATION OF LOAD STATEMENT IS BYPASSED FOR TABLE** *table-name*

Explanation: The PUNCHDDN specification is omitted in a UNLOAD statement when partition parallelism is activated. The LOAD statement will not be generated for the indicated table.

Severity: 0 (informational)

System action: The UNLOAD job will continue.

DSNU1204I DUPLICATE ROWS MIGHT HAVE BEEN UNLOADED

Explanation: The image copy contains duplicate pages. All rows from the duplicate pages are unloaded

Severity: 4 (warning)

System action: Processing continues.

DSNU1211I *csect-name* **- INPUT DATA SET IS NOT AN IMAGE COPY**

Explanation: The input data set specified by the FROMCOPY or FROMCOPYDDN option of the UNLOAD utility was not identified as an image copy data set.

Severity: 8 (error)

System action: The utility job terminates.

User response: Specify the correct image copy data set name to the FROMCOPY option, or to the DD-name that is referred to by the FROMCOPYDDN option.

DSNU1212I *csect-name* **- INVALID FIELD NAME** *field-name* **FOR TABLE** *table-name*

Explanation: The specified field name in the field specification list of the UNLOAD utility is:

- Not a column name of the specified table, or
- An existing column name for the CONSTANT option.

The field name is not valid.

Severity: 8 (error)

System action: The utility job terminates.

User response: For a CONSTANT field, specify a non-existing field name. Otherwise, specify the field name as a column name of the indicated table.

DSNU1213I *csect-name* **- INVALID FIELD TYPE FOR FIELD** *field-name* **FOR TABLE** *table-name*

Explanation: The specified field type in the field specification list of the UNLOAD utility is incompatible with the base type of the corresponding column of the table.

Severity: 8 (error)

System action: The utility job terminates.

User response: Specify only a compatible type allowed for the unload operation in the field specification list.

DSNU1214I *csect-name* **- INVALID FIELD POSITION SPECIFICATION FOR FIELD** *field-name*, **TABLE** *table-name*

Explanation: For a UNLOAD statement, specified field positions are not in ascending order in the field specification list of the indicated table.

Severity: 8 (error)

System action: The utility job terminates.

User response: Specify the field positions in the ascending order in the field specification list. If a certain order of the output fields is desired, change the order of the field in the list so that the field positions are in the ascending order.

DSNU1217I *csect-name* **- CONFLICTING FIELD POSITION AND/OR LENGTH SPECIFICATION FOR FIELD** *field-name*, **TABLE** *table-name*

Explanation: For a UNLOAD statement, the specified field position for the indicated field caused a conflict with the previous field. This is caused either by

- The position value specified for the field is too small.
- The length specification for the previous field is too large.

Severity: 8 (error)

System action: The utility job terminates.

User response: Specify the correct position or length parameters in the field specification list so that there are enough space for the output fields.

DSNU1218I *csect-name* **- LOGICAL RECORD LENGTH OF OUTPUT RECORD EXCEEDED THE LIMIT FOR TABLE** *table-name*

Explanation: The maximum length of the output records exceeded the physical limit for the UNLOAD utility.

Severity: 8 (error)

System action: The utility job terminates.

Problem determination: This problem will most likely occur when BLOB, CLOB, or DBCLOB columns are selected from a table to be unloaded.

User response: Consider one or more of the following options that can reduce the length of the output record, if possible:

- Use the NOPAD option.
- Unload only the necessary fields using the field specification options.
- Specify the LENGTH option with TRUNCATE and/or STRIP options for the applicable fields.

If the above options can not be applied, you will have to use an application to unload the data from the table.

DSNU1219I *csect-name* **- THE NUMBER OF RECORDS IN ERROR REACHED THE LIMIT** *nnnn*

Explanation: For the UNLOAD utility, the number of records in error reached the specified MAXERR limit.

Severity: 8 (error)

System action: The utility job terminates.

Problem determination: This problem will occur when the number of errors in the output record formatting, including data conversion errors, detected by the UNLOAD utility reached the limit specified by the MAXERR option. For each record in error, the corresponding message has been issued. The records in error will not be unloaded to the output data set. This message is issued only once, even though there can be a situation in which the number of individual error messages exceeds the specified MAXERR value. This situation occurs only when the UNLOAD utility is running in the partition parallel mode and the multiple tasks have reached the MAXERR value concurrently.

User response: Investigate the individual error message and determine the cause of the errors. The corrective action will differ by the type of the error. In certain situations, this condition may be anticipated. In this case, the total number of records unloaded before the job terminates can be adjusted by changing the MAXERR option value.

DSNU1220I *csect-name* **- UNLOADING LOB COLUMNS FROM IMAGE COPY IS NOT SUPPORTED FOR TABLE** *table-name*

Explanation: For the UNLOAD utility, one or more BLOB, CLOB, and/or DBCLOB columns are selected on the indicated table when the source object is an image copy data set.

Severity: 8 (error)

System action: The utility job terminates.

Problem determination: When the source object of the UNLOAD utility is an image copy, selection of a LOB column is not supported.

User response: Specify a field selection list that does not include any BLOB, CLOB, or DBCLOB column from the indicated table.

DSNU1221I *csect-name* **- UNLDDN IS NOT SPECIFIED OR SPECIFIED TEMPLATE** *template-name* **IS INVALID FOR MULTIPLE TABLE SPACES**

Explanation: For the UNLOAD utility, the output data set specification is invalid when the source objects consist of multiple table spaces.

Severity: 8 (error)

System action: The utility job terminates.

Problem determination: When the source objects of the UNLOAD utility consist of multiple table spaces, an output data set corresponding to each table space is required.

User response: Include an appropriate TEMPLATE statement in the unload job step that contains the table space name as a variable. If the table space is selected from different databases, the database name must also be included as a variable. Specify the template name to the UNLDDN option.

DSNU1222I *csect-name* **- TEMPLATE** *template-name* **SPECIFIED FOR PUNCHDDN IS INVALID FOR MULTIPLE TABLE SPACES**

Explanation: For the UNLOAD utility, the specification of output data set to store the generated LOAD statement is invalid when the source objects consist of multiple table spaces.

Severity: 8 (error)

System action: The utility job terminates.

Problem determination: When the source objects of the UNLOAD utility consist of multiple table spaces, an output data set corresponding to each table space is required.

User response: Include an appropriate TEMPLATE statement in the unload job step that contains the table space name as a variable. If the table space is selected from different databases, the database name must also be included as a variable. Specify the template name to the PUNCHDDN option. The TEMPLATE should be different from the one used for UNLDDN.

DSNU1223I *csect-name* **- TABLESPACE SPECIFICATION IS MISSING FOR FROMCOPY OR FROMCOPYDDN**

Explanation: For the UNLOAD utility, the FROMCOPY or FROMCOPYDDN option is used but the TEMPLATE specification is missing.

Severity: 8 (error)

System action: The utility job terminates.

Problem determination: When the source objects of the UNLOAD utility is an image copy, the TABLESPACE option is required to specify the table space name of the image copy.

User response: Add the TABLESPACE option in the UNLOAD statement to specify the table space name of the image copy.

DSNU1224I *csect-name* **- TABLE** *table-name* **IS NOT IN THE TABLE SPACE** *table-space-name*

Explanation: The indicated *table-name* given in a FROM TABLE option is not in the table space specified by the TABLESPACE option.

Severity: 8 (error)

System action: The utility job terminates.

Problem determination: There is an unmatch between the TABLESPACE and FROM TABLE specifications or the specified table was dropped. Dropped tables are not supported by the UNLOAD utility.

User response: Correct either the TABLESPACE or the FROM TABLE specification to select a table in the specified table space. If the table was dropped, remove the FROM TABLE option for the dropped table.

DSNU1225I *csect-name* **- TABLESPACE KEYWORD IS NOT SPECIFIED. AT LEAST ONE FROM TABLE SPECIFICATION IS REQUIRED.**

Explanation: There is no TABLESPACE nor FROM TABLE specification in a UNLOAD statement. The source object for unloading can not be identified.

Severity: 8 (error)

System action: The utility job terminates.

User response: If the TABLESPACE specification is omitted, specify at least one FROM TABLE option to identify the source object. Instead of a TABLESPACE specification, the LIST option can be used, if a LISTDEF statement defines a list of objects. When the LIST option is used, the FROM TABLE option can not be specified.

DSNU1226I *csect-name* **- THE NUMBER OF OBJECTS IS TOO LARGE**

Explanation: The number of source objects for the UNLOAD utility is too large. Overflow in some of the internal data structures occurred.

Severity: 8 (error)

Problem determination: This problem will most likely occur when the LIST option is used and the corresponding LISTDEF definition is expanded to a large number of objects.

System action: The utility job terminates.

User response: Divide the UNLOAD job step into two or more job steps and, in each job step, use a LISTDEF that expands to less number of objects.

DSNU1227I *csect-name* **- THE CONTENT OF THE IMAGE COPY DATA SET IS INCONSISTENT WITH TABLE SPACE** *table-space*

Explanation: For the UNLOAD utility, the data set specified by the FROMCOPY or the FROMCOPYDDN option is not corresponding to the table space.

Severity: 8 (error)

System action: The utility job terminates.

User response: Specify a correct image copy data set name for the table space, or specify the correct table space name for the image copy.

DSNU1228I *csect-name* **- A ROWID COLUMN** *column-name*, **TABLE** *table-name* **IS SELECTED FOR UNLOAD BUT THE PAGE SET HEADER PAGE IS MISSING IN THE SPECIFIED IMAGE COPY DATA SET**

Explanation: For the UNLOAD utility, a ROWID column is selected for unloading from an image copy, but the required information to form the ROWID can not be retrieved because the page set header page is not included in the specified image copy data set.

Severity: 8 (error)

System action: The utility job terminates.

Problem determination: The problem will most likely occur when a piece of copy data set, not the first one, is

specified to the FROMCOPY option.

User response: Use the FROMCOPYDDN, instead of FROMCOPY, to specify a DD name where the first piece of copy that contain the page set header page is concatenated to the source image copy data set.

DSNU1229I *csect-name* **- ERROR IN LOADING DICTIONARY FOR TABLE SPACE** *table-space*

Explanation: For the UNLOAD utility, an error was detected in loading the dictionary for decompression while reading an image copy data set.

Severity: 8 (error)

System action: The utility job terminates.

Problem determination: Possibly the image copy data set is corrupted.

User response: If the right image copy data set is specified to the FROMCOPY or FROMCOPYDDN, it will be required to take a new image copy for the table space by running the COPY utility or using other alternative methods.

DSNU1230I *csect-name* **- INCONSISTENT STRIP CHAR LENGTH FOR FIELD** *field-name*, **TABLE** *table-name*

Explanation: An invalid strip character was specified for the indicated field of the table.

Severity: 8 (error)

System action: The utility job terminates.

Problem determination: For an SBCS or MIXED character field, the strip character, if specified, must be an SBCS character; for a DBCS field, it must be a DBCS character.

User response: Specify the correct strip character applicable to the field.

DSNU1231I *csect-name* **- LENGTH OF OUTPUT RECORD IS TOO LONG FOR TABLE** *table-name*

Explanation: An output record for the indicated table exceeded the physical limit.

Severity: 4 (warning)

System action: The record in error will not be unloaded to the output data set. The UNLOAD utility continues until the total number of records in error reaches the value specified by the MAXERR option (the default is 1).

If the total number of records reaches the MAXERR value, the UNLOAD utility issues the DSNU1219I message and terminates with the return code 8.

Problem determination: This message is issued only when the NOPAD option is specified in the UNLOAD statement and most likely when BLOB, CLOB, or DBCLOB columns are selected for unload from the table. It is also possible that, if the expansion of output strings occurs due to the encoding scheme specified for the output, the total length of some output records becomes longer than expected.

User response: Consider the following options to reduce the size of the output records:

- Unload only the necessary fields using the field specification options.
- Specify the LENGTH option with TRUNCATE and/or STRIP options for the applicable fields.

If the above options can not be applied, you will have to use an application to unload the data from the table.

DSNU1232I *csect-name* **- COMPRESSED ROW IS IGNORED BECAUSE THE DICTIONARY IS NOT AVAILABLE FOR TABLE** *table-name*

Explanation: The UNLOAD utility encountered a compressed row when unloading the data from an image copy data set but the dictionary for decompression has not been loaded for the table space.

Severity: 4 (warning)

System action: The record in error will not be unloaded to the output data set. The UNLOAD utility continues until the total number of records in error reaches the value specified by the MAXERR option (the default is 1). If the total number of records reaches the MAXERR value, the UNLOAD utility issues the DSNU1219I message and terminates with the return code 8.

Problem determination: This message is issued when a copy of pieces (not the first piece) of a compressed table space is specified in the FROMCOPY or FROMCOPYDDN option of the UNLOAD utility.

User response: To unload compressed rows from the copy of pieces, use COPYDDN option to specify a DD name under which the first copy of pieces that contains the compression dictionary is concatenated.

DSNU1233I *csect-name* **- DATA IS TOO LONG FOR FIELD** *field-name*, **TABLE** *table-name*

Explanation: The space for the field in an output record is not enough to unload the data from the corresponding column of a row in the indicated table.

Severity: 4 (warning)

System action: The record in error will not be unloaded to the output data set. The UNLOAD utility continues until the total number of records in error reaches the value specified by the MAXERR option (the default is 1).

If the total number of records reaches the MAXERR value, the UNLOAD utility issues the DSNU1219I message and terminates with the return code 8.

Problem determination: This message is issued when the space of a field in an output record is limited by the LENGTH specification for the field or the POSITION specification for the next field.

User response: Apply appropriate values to the LENGTH and/or POSITION parameters in the field specification list to accommodate the data in the output records. Alternatively, specify TRUNCATE option to the field, if applicable. Specifying the STRIP option, if applicable, can also reduce the length of data if the data contains leading or trailing blanks.

DSNU1234I *csect-name* **- NUMERIC CONVERSION ERROR OCCURRED FOR FIELD** *field-name*, **TABLE** *table-name*

Explanation: The UNLOAD utility encountered an error when it tried to convert a data of one numeric type to another.

Severity: 4 (warning)

System action: The record in error will not be unloaded to the output data set. The UNLOAD utility continues until the total number of records in error reaches the value specified by the MAXERR option (the default is 1).

If the total number of records reaches the MAXERR value, the UNLOAD utility issues the DSNU1219I message and terminates with the return code 8.

Problem determination: This message is issued most likely when an overflow occurs in converting a data of one numeric type to another. For example, an attempt to convert an INTEGER 32768 to a SMALL INT causes an overflow.

User response: To eliminate any numeric overflow error, specify the same numeric data type for the output as the base type of the columns in the table. Convert numeric types only when it is known that the range of the source data fits in the allowable range of the specified numeric type for the output.

DSNU1235I *csect-name* **- CCSID SUBSTITUTION IS ATTEMPTED WITH NOSUBS OPTION FOR FIELD** *field-name*, **TABLE** *table-name*

Explanation: While the UNLOAD utility is running with the NOSUBS option, a CCSID substitution is attempted in the CCSID translation process for the indicated field.

Severity: 4 (warning)

System action: The record in error will not be unloaded to the output data set. The UNLOAD utility continues until the total number of records in error reaches the value specified by the MAXERR option (the default is 1). If the total number of records reaches the MAXERR value, the UNLOAD utility issues the DSNU1219I message and terminates with the return code 8.

User response: If it is desired to eliminate any CCSID substitution, specify an appropriate CCSID that has the corresponding entries to the original CCSID, if available. To allow the default CCSID substitution, remove the NOSUBS option.

DSNU1236I *csect-name* **- POSITION SPECIFICATION CAUSES FIELD OVERLAP FOR FIELD** *field-name*, **TABLE** *table-name*

Explanation: For the UNLOAD utility, the space for an output field was overlapped by the indicated field to which the POSITION parameter is specified.

Severity: 4 (warning)

System action: The record in error will not be unloaded to the output data set. The UNLOAD utility continues until the total number of records in error reaches the value specified by the MAXERR option (the default is 1). If the total number of records reaches the MAXERR value, the UNLOAD utility issues the DSNU1219I message and terminates with the return code 8.

Problem determination: This message will be issued only when the NOPAD option is specified in the UNLOAD statement. The POSITION specification for the field caused a conflict with the positioning and/or the length of a previous field.

User response: Consider the following options to eliminate this situation:

- Remove the POSITION specifications from the field specification list.
- Adjust the POSITION parameter for the field to preserve enough space for the previous fields.
- Adjust the LENGTH, possibly with the TRUNCATE option, of the previous fields, if applicable.

DSNU1237I *csect-name* **- ERROR IN CCSID CONVERSION FOR FIELD** *field-name*, **TABLE** *table-name*

Explanation: The CCSID translation process detected a code point or form exception while the UNLOAD utility was running.

Severity: 4 (warning)

System action: The record in error will not be unloaded to the output data set. The UNLOAD utility continues until the total number of records in error reaches the value specified by the MAXERR option (the default is 1). If the total number of records reaches the MAXERR value, the UNLOAD utility issues the DSNU1219I message and terminates with the return code 8.

User response: If the specified CCSID for output is incorrect, correct the CCSID specification. If a valid CCSID is specified, notify the system programmer.

System programmer response: Verify the contents of SYSSTRINGS. If necessary, insert a row into SYSSTRINGS for the translation. For an SBCS translation, the row must contain a translate table or the name of a user-provided translate procedure. For a DBCS translation, the row must contain the name of a user-provided translate procedure.

DSNU1238I TABLE *table2* **IS NOT IN SAME TABLESPACE AS TABLE** *table1*

Explanation: All source tables for a single UNLOAD utility statement must be in the same table space.

Severity: 8 (error)

System action: Utility processing terminates.

User response: Remove the FROM TABLE control statement that specifies *table2* from the current UNLOAD request. Create another UNLOAD utility statement that contains the FROM TABLE control statement for *table2* so that all source tables in a single UNLOAD request are from the same table space. Resubmit the job.

DSNU1241I *csect-name*- *utility-name* **UTILITY CANNOT PROCEED BECAUSE TABLE** *tablename* **IS SPECIFIED IN MORE THAN ONE FROM TABLE SPECIFICATION**

Explanation: The table *table-name* cannot be specified in multiple FROM TABLE specifications.

Severity: 8 (error)

System action: Utility processing terminates.

User response: Remove the FROM TABLE specifications for the same table. Group all WHEN control statements for the same table into a single WHEN control statement by using the OR keyword. Resubmit the job.

DSNU1250I *csect-name* **- ERROR IN CCSID CONVERSION FOR** *object-type***. SOURCE CCSID=***ccsid***, TARGET CCSID=***ccsid***

Explanation: The CCSID translation process failed for *object-type*. The values of *object-type* and the cause of the error for each type are:

- SBCS PAD CHAR
- DBCS PAD CHAR
- VERIFICATION
- HEADER CONST
- *field-name*

Severity: 8 (error)

System action: The utility terminates.

User response: If the CCSID that is specified in the utility control statement is incorrect, correct the CCSID specification. If a valid CCSID is specified, notify the system programmer.

System programmer response: Check the SYSSTRINGS catalog table to determine whether a row exists for the CCSID conversion. If a row does not exist, and OS/390 V2R9 is installed, refer to *z/OS C/C++ Programming Guide* for additional conversions that are supported. If no conversions are available, contact your IBM service representative to request support for the conversion.

DSNU1251I *csect-name* **- ERROR IN CCSID CONVERSION FOR** *object-type* **IN FIELD** *field-name* **OF TABLE** *table-name***. SOURCE CCSID=***ccsid***, TARGET CCSID=***ccsid***

Explanation: The CCSID translation process failed for *object-type* . The values of *object-type* and the cause of the error for each type are:

- STRIP CHAR

Severity: 8 (error)

System action: The utility terminates.

System programmer response: Check the SYSSTRINGS catalog table to determine whether a row exists for the CCSID conversion. If a row does not exist, and OS/390 V2R9 is installed, refer to *z/OS C/C++ Programming Guide* for additional conversions that are supported. If no conversions are available, contact your IBM service representative to request support for the conversion.

User response: If the CCSID that is specified in the utility control statement is incorrect, correct the CCSID specification. If a valid CCSID is specified, notify the system programmer.

DSNU1252I THE PARTITION VARIABLE IN THE TEMPLATE DSN WAS SPECIFIED BUT NO PARTITION IS ACTIVATED

Explanation: A utility control statement for the UNLOAD utility contains a template name in the UNLDDN option. The template contains a variable that represents a partition number. However, the source object is either a non-partitioned table space or an image copy that does not support partition parallelism.

Severity: 4 (warning)

System action: UNLOAD processing continues. For the indicated table space, a single output data set is allocated, and the unloaded data is placed in that data set.

DSNU1253I *csect-name* **- USER** *user-id* **DOES NOT HAVE THE SELECT PRIVILEGE ON TABLE** *table-name* **IN TABLE SPACE** *table-space-name*

Explanation: User *user-id* attempted to run a utility that reqires the SELECT privilege on table *table-name*. The user does not have that privilege.

Severity: 8 (error)

System action: The utility terminates.

System programmer response: Grante the user the SELECT privilege on table *table-name* in table space *tabel-space-name*.

DSNU1254I *csect-name* **FORMAT DELIMITED REQUIRES ONE AND ONLY ONE FROM TABLE**

Explanation: UNLOAD FORMAT DELIMITED supports unloading only from a single table. FORMAT DELIMITED was specified either with:

- Multiple FROM TABLE specifications, or
- No FROM TABLE specifications, with a table space that contains more than one table.

Severity: 8 (error)

System action: Processing terminates.

User response: Correct the UNLOAD statement and resubmit the utility.

DSNU1300I *csect-name* **MODIFY STATISTICS COMPLETED SUCCESSFULLY**

Explanation: This message reports the successful completion of the MODIFY STATISTICS utility operation.

Severity: 0 (informational)

DSNU1301I *csect-name n* **SYSIBM.SYSCOLDIST_HIST ROWS WERE DELETED**

Explanation: This message displays the number *n* of SYSIBM.SYSCOLDIST_HIST rows that were deleted by the MODIFY STATISTICS utility.

Severity: 0 (informational)

DSNU1302I *csect-name n* **SYSIBM.SYSCOLUMNS_HIST ROWS WERE DELETED**

Explanation: This message displays the number *n* of SYSIBM.SYSCOLUMNS_HIST rows that were deleted by the MODIFY STATISTICS utility. The rows were deleted based on the AGE criteria or the DATE criteria that the user specified.

Severity: 0 (informational)

DSNU1303I *csect-name n* **SYSIBM.SYSINDEXES_HIST ROWS WERE DELETED**

Explanation: This message displays the number *n* of SYSIBMS.SYSINDEXES rows that were deleted by the MODIFY STATISTICS utility. The rows were deleted based on the AGE criteria or the DATE criteria that the user specified.

Severity: 0 (informational)

DSNU1304I *csect-name n* **SYSIBM.SYSINDEXPART_HIST ROWS WERE DELETED**

Explanation: This message displays the number *n* of SYSIBM.SYSINDEXPART_HIST rows that were deleted by the MODIFY STATISTICS utility. The rows were deleted due to the AGE criteria or the DATE criteria that the user specified.

Severity: 0 (informational)

DSNU1305I *csect-name n* **SYSIBM.SYSINDEXSTATS_HIST ROWS WERE DELETED**

Explanation: This message displays the number *n* of SYSIBM.SYSINDEXSTATS_HIST rows that were deleted by the MODIFY STATISTICS utility. The rows were deleted due to the AGE criteria or the DATE criteria that the user specified.

Severity: 0 (informational)

DSNU1306I *csect-name n* **SYSIBM.SYSLOBSTATS_HIST ROWS WERE DELETED**

Explanation: This message displays the number *n* of SYSIBM.SYSLOBSTATS_HIST rows that were deleted by the MODIFY STATISTICS utility. The rows were deleted due to the AGE criteria or the DATE criteria that the user specified.

Severity: 0 (informational)

DSNU1307I *csect-name n* **SYSIBM.SYSTABLEPARTS_HIST ROWS WERE DELETED**

Explanation: This message displays the number *n* of SYSIBM.SYSTABLEPARTS_HIST rows that were deleted by the MODIFY STATISTICS utility. The rows were deleted due to the AGE criteria or the DATE criteria that the user specified.

Severity: 0 (informational)

DSNU1308I *csect-name n* **SYSIBM.SYSTABLES_HIST ROWS WERE DELETED**

Explanation: This message displays the number of SYSIBM.SYSTABLES_HIST rows that were deleted by the MODIFY STATISTICS utility. The rows were deleted due to the AGE criteria or the DATE criteria that the user specified.

Severity: 0 (informational)

DSNU1309I *csect-name n* **SYSIBM.SYSTABSTATS_HIST ROWS WERE DELETED**

Explanation: This message displays the number *n* of SYSIBM.SYSTABSTATS_HIST rows that were deleted by the MODIFY STATISTICS utility. The rows were deleted due to the AGE criteria or the DATE criteria that the user specified.

Severity: 0 (informational)

DSNU1310I **THE IMAGE COPY FOR** *obj-type dbname.tsname* **DSNUM** *dsnum* **RESIDES ON A TAPE VOLUME BUT IS NOT PREALLOCATED VIA JCL AND TAPEUNITS IS 0**

Explanation: The specification of TAPEUNITS=0 restricts the utility from using tape units that are not preallocated via JCL.

Severity: 8 (error)

System action: Processing stops for this object.

User response: Either change the value on the TAPEUNITS keyword and resubmit the job, or preallocate the image copy and resubmit the job.

DSNU1400I *csect-name* **- COPYTOCOPY UTILITY NOT SUPPORTED FOR** *obj-type obj-qual.obj-name* **DSNUM** *n*

Explanation: COPYTOCOPY cannot make copies of:

- DSNDB01.SYSUTILX, DSNDB01.DBD01, or DSNDB06.SYSCOPY table spaces, or any of their indexes.
- An image copy that was taken using the COPY utility with the CONCURRENT option.

Severity: 8 (error)

System action: Processing halts for this object. If the utility job step includes an OPTIONS control statement with the EVENT(ITEMERROR,SKIP) option, COPYTOCOPY processes the next item in the list.

User response: If the unsupported object is a restricted catalog or directory object listed in the explanation, use another method to make all image copies for this object.

If the unsupported object is an image copy made with the CONCURRENT option, then make a primary local image copy by running the COPY utility without the CONCURRENT option. Rerun COPYTOCOPY to make additional image copies.

DSNU1401I *csect-name* **- UNABLE TO LOCATE IMAGE COPY FOR** *obj-type obj-qual.obj-name* **DSNUM** *n*

Explanation: One of the following conditions prompted this error code:

- The COPYTOCOPY utility attempted to make additional image copies from the referenced object. However, SYSIBM.SYSCOPY does not contain a record of the referenced object.
- UNLOAD utility could not find a record in SYSIBM.SYSCOPY for the image copy dataset that was requested.

Severity: 8 (error)

System action: Processing halts for this object.

User response: Make a primary local image copy by running the COPY utility, then run COPYTOCOPY to make additional image copies.

DSNU1402I *csect-name* **-** *copy-type* **FOR** *obj-type obj-qual.obj-name* **DSNUM** *n* **WITH START_RBA** *rba* **ALREADY EXISTS**

Explanation: COPYTOCOPY attempted to make an image copy that already exists.

Severity: 4 (warning) or 8 (error)

System action: The utility does not make the image copy for the specified object type. COPYTOCOPY makes the remaining types of image copies. If there is a list of objects to be copied, COPYTOCOPY processes the next object in the list.

User response: Run the REPORT utility to determine the image copy types that you need. Modify your COPYTOCOPY job, and rerun it as needed.

DSNU1403I *csect-name* **-** *copy-type* **DATA SET** *dsn* **WITH START_RBA** *rba* **IS IN USE BY COPYTOCOPY FOR** *obj-type obj-qual.obj-name* **DSNUM** *n*

Explanation: The COPYTOCOPY utility issues this message before it opens an image copy data set and begins the copying process.

Severity: 0 (informational)

System action: COPYTOCOPY processing continues.

DSNU1404I *csect-name* **- COPYTOCOPY PROCESSING COMPLETED FOR** *obj-type obj-qual.obj-name* **DSNUM** *n* **ELAPSED TIME =** *elapsed time* **NUMBER OF PAGES COPIED =** *n*

Explanation: COPYTOCOPY issues this message at the normal completion of an object. The time that elapsed since the beginning of processing the object to its completion was *hh* hours, *mm* minutes, and *ss* seconds.

Severity: 0 (informational)

System action: The COPYTOCOPY utility completed normally for this object. Processing continues.

DSNU1405I *csect-name* **-** *site-type* **SITE PRIMARY IMAGE COPY DOES NOT EXIST FOR** *obj-type obj-qual.obj-name* **DSNUM** *n*

Explanation: The COPYTOCOPY utility attempted to make a backup image copy for the site, but the primary image copy does not exist.

Severity: 8 (error)

System action: The utility cannot make the site backup copy. If the utility job includes an OPTIONS control statement with the EVENT(ITEMERROR,SKIP) option, COPYTOCOPY processes the next item on the list.

User response: Modify your COPYTOCOPY job, COPYDDN or RECOVERYDDN option, to include the ddname for the primary copy.

DSNU1406I *csect-name* **- COPYTOCOPY COMPLETED. ELAPSED TIME =** *elapsed_time*

Explanation: DB2 issues this message at the normal completion of the COPYTOCOPY utility. The time that elapsed since the beginning of the COPYTOCOPY utility to its completion was *hh* hours, *mm* minutes, and *ss* seconds.

Severity: 0 (informational)

System action: Processing continues to the next utility statement in the utility job step, if any. If not, the utility job step halts, and the UTILID of the utility job step becomes available for reuse.

DSNU1407I *csect-name* **- MISMATCH BETWEEN THE NUMBER OF PAGES READ=***count* **AND THE NUMBER OF PAGES ORIGINALLY COPIED=***count* **FOR INPUT DATA SET** *dsname* **OBJECT NAME** *obj-type obj-qual.obj-name* **DSNUM** *n*

Explanation: The COPYTOCOPY or RECOVER utility issues this message when it detects that the number of pages read from the input image copy does not match the original number of pages copied by the COPY utility. The utility retrieves the number of pages originally copied from the COPYPAGESF column of the SYSIBM.SYSCOPY record for the input image copy data set.

Severity: 8 (error)

System action: The COPYTOCOPY utility terminates processing of this object. The RECOVER utility attempts to continue by falling back to the previous full image copy, if any. If the utility job step includes an OPTIONS control statement with the EVENT(ITEMERROR,SKIP) option, COPYTOCOPY processes the next item in the list.

User response: Report to IBM Center. Provide output of COPYTOCOPY job, the data set referenced in the message, and run REPORT utility on the object and provide its output.

DSNU1408I *csect-name* **- SOURCE DATA SET** *dsname* **DOES NOT MATCH** *obj-type obj-qual.obj-name* **DSNUM** *n*

Explanation: The COPYTOCOPY or RECOVER utility issues this message when the DBID or PSID of the source data set does not match the DBID or PSID of the table space or index space for which the copy is made. This error can occur if the data set was reused.

The COPY utility displays this message when the DBID or PSID of the header page of the target object does not match the information in the DBD. In this case, the source data set dsname will be omitted from the message.

Severity: 4 (warning) or 8 (error)

System action: RECOVER attempts to continue processing by falling back to the previous full image copy, if any.

COPYTOCOPY searches for the next available image copy data set with the same START_RBA value as the START_RBA value registered in SYSIBM.SYSCOPY for dsname. If COPYTOCOPY finds a data set, it uses that data set. Otherwise, COPYTOCOPY issues message DSNU1401I.

If FROMCOPY is specified, processing of this object halts.

COPY fails with RC8, and the table space is placed in the RECOVER PENDING state.

User response: Determine why the data set was reused for the image copy data set that is registered in SYSIBM.SYSCOPY.

If FROMCOPY is specified, correct the COPYTOCOPY utility control statement. Specify a source data set that is an image copy of the table space or index space for which you want additional image copies.

If this message was received for COPY, try to
determine how the header page of the object became
corrupted, and fix it using the REPAIR utility, if
possible. Otherwise, recover the object using a good
copy.

DSNU1409I *csect-name* **-** *ttttt* **ERRORS ON DATA SET NAME=***dsname obj-type obj-qual.obj-name* **DSNUM** *n*

Explanation: DB2 detected I/O errors on the named object. *ttttt* is either 'READ' or 'WRITE'.

Severity: 4 (warning)

System action: If FROMCOPY is specified, processing is terminated. Otherwise, processing will continue. The utility will attempt to use the next available image copy data set that matches COPYTOCOPY job specification with the same START_ RBA as the data set named in error. If it fails to find one, processing of this object is terminated.

User response: There may be an IOS message on the MVS console SYSLOG indicating the type of I/O error that occurred. If an IOS message was issued, follow the Problem Determination directions for the message.

DSNU1410I *csect-name* **- INCREMENTAL IMAGE COPY DISALLOWED FOR** *obj-type obj-qual.obj-name* **DSNUM** *n***. FULL IMAGE COPY WILL BE USED**

Explanation: An incremental image copy is disallowed for the specified object type. The utility ignores the request, and creates a full image copy if available.

Severity: 4 (warning)

System action: The system treats the request to copy an incremental image copy as a request to copy a full image copy.

User response: The capacity of the target image copy data set may not be sufficient if its size was established assuming an incremental image copy. If this condition occus, halt the utility and resubmit the job step with the appropriate size.

DSNU1411I *csect-name* **-** *ut_name* **DOES NOT SUPPORT RESTART(PHASE). PHASE CHANGED TO CURRENT.**

Explanation: This utility does not support the RESTART(PHASE).

Severity: 0 (informational)

System action: The utility overrides RESTART(PHASE) with RESTART(CURRENT).

DSNU1450I *csect-name* **- CHARACTER CONVERSION FROM CCSID***from-ccsid* **TO CCSID** *to-ccsid* **FAILED WITH ERROR CODE 'error-code' FOR TABLE** *dbid.obid* **COLUMN 'column-number'.**

Explanation: A conversion error occurred during the conversion of a string to a different coded character set. The error occurred during the processing of the utility statement.

The *from-ccsid* identifies the coded character set of to be converted. and the *to-ccsid* identifies the coded character set to which it must be converted.

The 'error-code' indicates the type of error as defined as follows:

8 Length exception (for example, expansion required for PC MIXED data exceeds the maximum length of the string).

12 Invalid code point

16 Form exception (for example, invalid MIXED data).

20 Translate procedure error (for example, an error during Unicode/390 stage 2 conversions or an exit set the length control field of the string to an invalid value).

24 the conversion services have not been setup correctly or the conversion is not supported.

The *dbid.obid* gives the database id and the table obid of the object with the conversion error. This could be either the target or the source of the conversion.

The 'column-number' gives the column number of the failing object. This could be either the target or the source of the conversion.

Severity: 8 (error)

System action: The utility job is terminated.

User response: Refer to Part 2 of DB2 Diagnosis Guide and Reference for information on identifying and reporting the problem. The The user may generate a dump by adding the DIAGNOSE ABEND statement in the utility job and re-running the utility. If the error can be fixed, correct the error and resubmit the utility, else issue the TERM utility command to terminate the utility.

DSNU1502I *csect-name* **- RECOVER OF** *obj-type obj-qual.objname dsnum n* **WILL PROCEED TO LOG POINT X'***rba/lrsn***'**

Explanation: DB2 issues this message if the RECOVER utility statement includes the TOLASTCOPY or TOLASTFULLCOPY keyword. The message provides information on the point in time that the object was recovered to. In a data sharing environment, the log

point identified is an LRSN. Otherwise, the log point identified is an RBA.

Severity: 0 (informational)

System action: The RECOVER utility statement continues processing.

DSNU1510I *csect-name* **- LOG APPLY PHASE COMPLETE, ELAPSED TIME:** *HH.MM.SS*

Explanation: This message is issued at the normal completion of the LOG APPLY phase of the RECOVER utility. The time that has elapsed between the beginning and completion of this phase was:

hh hours

mm minutes

ss seconds

Severity: 0 (informational)

System action: Processing continues.

DSNU1511I *csect-name* **- FAST LOG APPLY WAS NOT USED FOR RECOVERY**

Explanation: The RECOVER utility issues this message to notify the user that FAST LOG APPLY was not used. FAST LOG APPLY is not used for the following reasons:

- The ZPARM DSN6SYSP LOGAPSTG has 0 or insufficient storage specified.
- The storage that is used by currently active RECOVER jobs exceeds what was specified in the DSN6SYSP LOGAPSTG ZPARM.

Severity: 0 (informational)

System action: The RECOVER utility continues processing.

DSNU1521I *csect-name* **A LOGONLY RECOVERY CANNOT BE DONE BECAUSE THE HEADER PAGE CANNOT BE ACCESSED FOR** *obj-type obj-qual.obj-name dsnum n*

Explanation: The header page could not be read. DB2 stops the RECOVER utility from proceeding with a LOGONLY recovery because the starting point is unavailable.

Severity: 8 (error)

System programmer response: REMOVE the LOGONLY keyword from the RECOVER utility statement and recover the table space by using an image copy along with a log.

DSNU1600I *csect-name* **- BACKUP SYSTEM UTILITY FOR** *object* **STARTING, COPY POOL =** *copy-pool-name*, **TOKEN = X'***token***'**

Explanation: Backup of the indicated object, either data or logs, to the indicated copy pool and with the indicated token is starting.

Severity: 0 (informational)

System action: Backup of the indicated object has started.

System programmer response: None.

DSNU1601I *csect-name* **- BACKUP SYSTEM UTILITY IS NOT RESTARTABLE. RESTART IGNORED. A NEW BACKUP WILL BE INITIATED**

Explanation: The BACKUP SYSTEM utility cannot be restarted. The restart is ignored and a new backup is done.

Severity: 0 (informational)

System action: The restart is ignored, and a new BACKUP SYSTEM request is initiated.

System programmer response: None.

DSNU1602I *csect-name* **- BACKUP SYSTEM UTILITY COMPLETED, ELAPSED TIME =** *hh:mm:ss*

Explanation: The BACKUP SYSTEM utility cannot be restarted. The restart is ignored, and a new backup is done.

Severity: 0 (informational)

System action: None.

System programmer response: None.

DSNU1603I *csect-name* **- RESTORE SYSTEM UTILITY RESTART FAILED BECAUSE THE MEMBER-ID,***member-id***, ON WHICH THE RESTART HAS BEEN ISSUED IS DIFFERENT FROM THE MEMBER-ID,** *member-id***, ON WHICH THE RESTORE WAS ORIGINALLY DONE**

Explanation: A group RESTORE in a data sharing environment is already in the log apply phase. The RESTORE was originally issued and owned by a member other than the one that issued the restart, which is not allowed.

Severity: 8 (error)

System action: The restart fails.

System programmer response: Issue the restart job from the member that was indicated in the message.

DSNU1604I *csect-name* **- RESTORE SYSTEM PHASE LOG APPLY STARTED AT LOG POINT =** *log-point-value*

Explanation: The RESTORE SYSTEM utility has begun applying logs beginning at the log-point-value in order to arrive at the point in time indicated by the log point at which the logs were truncated via the SYSPITR parameter on the conditional restart command, CRESTART CREATE.

Severity: 0 (informational)

System programmer response: None.

DSNU1605I *csect-name* **- BACKUP SYSTEM UTILITY FAILED BECAUSE DB2 IS NOT IN VERSION 8 NEW FUNCTION MODE OR LATER**

Explanation: The BACKUP SYSTEM utility can be run only when DB2 is in VERSION 8 'new function mode' or later.

Severity: 8 (error)

System action: The utility is terminated.

System programmer response: Complete the conversion of DB2 to 'new function mode' before attempting to run the BACKUP SYSTEM utility.

DSNU1606I *csect-name* **- RESTORE SYSTEM UTILITY STARTING, COPY POOL =** *copy-pool-name*, **TOKEN = X'***token***'**

Explanation: Restore of the indicated copy pool with the indicated token is starting.

Severity: 0 (informational)

System action: None.

System programmer response: None.

DSNU1607I *csect-name* **- RESTORE SYSTEM UTILITY REQUIRES INSTALL SYSADM AUTHORITY TO RUN. THE CALLER HAS INSUFFICIENT AUTHORITY**

Explanation: The invoker of the RESTORE SYSTEM utility lacks the install SYSADM authority required to run the utility.

Severity: 8 (error)

System action: The utility is terminated.

System programmer response: Ensure that the RESTORE SYSTEM utility is run by an administrator with install SYSADM authority.

DSNU1608I *csect-name* **-** *system-utility* **SYSTEM UTILITY FAILED. REQUIRED DFSMSHSM SERVICES NOT AVAILABLE.**

Explanation: The system that attempted to run the BACKUP SYSTEM or RESTORE SYSTEM utility does not have the version of z/OS that contains the version of DFSMSHSM that is required by those utilities.

Severity: 8 (error)

System action: The utility is terminated.

System programmer response: Ensure that the required version of the z/OS operating system is installed on the system on which the BACKUP SYSTEM and RESTORE SYSTEM utilities are to be run.

DSNU1609I *csect-name* **- BACKUP SYSTEM UTILITY REQUIRES SYSADM OR SYSCTRL AUTHORITY TO RUN. THE CALLER HAS INSUFFICIENT AUTHORITY**

Explanation: The invoker of the BACKUP SYSTEM utility lacks the SYSADM or SYSCTRL authority required to run the utility.

Severity: 8 (error)

System action: The utility is terminated.

System programmer response: Ensure that the BACKUP SYSTEM utility is run by an administrator with SYSADM or SYSCTRL authority.

DSNU1610I *csect-name* **- AN INCOMPLETE BACKUP SYSTEM HISTORY ENTRY HAS BEEN FOUND IN THE BSDS, INDICATING THERE MAY BE AN INVALID DATA AND LOG VERSION IN COPYPOOL =** *copy-pool-name* **FOR TOKEN = X'***token***'. DFSMSHSM WILL BE CALLED TO DELETE THE VERSION**

Explanation: During Backup System processing, an incomplete backup history entry was found in the BSDS. For example, an incomplete entry can exist if DB2 or z/OS goes down before backup processing can complete. Such an entry indicates that a data and log version may exist in the indicated copy pool for the indicated token.

Severity: 0 (informational)

System action: Backup System will invoke DFSMShsm to delete the version of the data and logs before proceeding with the current backup.

System programmer response: None.

DSNU1611I *csect-name* **- A FAILURE HAS OCCURRED DURING BACKUP SYSTEM PROCESSING. DFSMSHSM IS BEING CALLED TO DELETE THE INCOMPLETE VERSION IN COPYPOOL =** *copy-pool-name* **FOR TOKEN = X'***token***'**

Explanation: A failure that was previously indicated during Backup System processing has created an incomplete version. The copy pool and token with the incomplete version are indicated in the message text. This incomplete version must be deleted.

Severity: 0 (informational)

System action: Backup System will invoke DFSMShsm to delete the incomplete version.

System programmer response: None.

DSNU1612I *csect-name* **- RESTORE SYSTEM UTILITY FAILED BECAUSE THE SYSTEM IS NOT IN SYSTEM RECOVER PENDING STATE.**

Explanation: The RESTORE SYSTEM utility cannot be run unless the proper preparation has been made. That preparation includes truncation of logs and performing a conditional restart of the appropriate system(s) so as to place them in the System RECOVER PENDING state. This procedure is detailed in the Part 4 (Volume 1) of *DB2 Administration Guide* under 'System point in time recovery'.

Severity: 8 (error)

System action: The utility is terminated.

System programmer response: Review the procedures for preparing systems for running the RESTORE SYSTEM utility found in Part 4 (Volume 1) of *DB2 Administration Guide* under 'System point in time recovery'.

DSNU1614I *csect-name* **- BACKUP SYSTEM UTILITY FOR** *object* **COMPLETED SUCCESSFULLY, COPY POOL =** *copy-pool-name***, TOKEN = X'***token***', ELAPSED TIME =** *hh:mm:ss*

Explanation: The backup of the indicated object, data or logs, has completed successfully.

Severity: 0 (informational)

System action: None.

System programmer response: None.

DSNU1616I *csect-name* **- THE CALL TO DFSMShsm TO RESTORE SYSTEM FAILED. PLEASE SEE MESSAGE ARC1806E IN THE OUTPUT FOR DETAILS. COPY POOL =** *copy-pool-name***, TOKEN = X'***token***'. A RETRY WILL BE ATTEMPTED IF AN APPROPRIATE BACKUP VERSION CAN BE IDENTIFIED.**

Explanation: The call to DFSMShsm to RESTORE SYSTEM with the indicated copy pool name and token failed during DFSMShsm processing. DFSMSHSM will have issued message ARC1806E indicating the cause of the error. This message can be found in the job log of the RESTORE SYSTEM utility.

If this is the first attempt by the RESTORE SYSTEM utility to do the RESTORE, a second attempt will be made if there is another appropriate Backup version recorded in the BACKUP SYSTEM history BSDS record. An appropriate Backup version is one for which data backup completed before the log truncation point indicated in the SYSPITR parameter of the conditional restart command, CRESTART CREATE. Messages indicating the progress of a subsequent attempt will be issued, just as for the first attempt.

If the second attempt fails, no further attempts will be made and the request will be failed.

Severity: 8 (error)

System action: If this is the first attempt to do the RESTORE and there is another appropriate Backup available, a second attempt will be made. Otherwise the utility will terminate.

System programmer response: Correct the problem identified in message ARC1806E and rerun the utility.

Problem determination: The job log will contain the ARC1806E message that is issued by DFSMShsm. This message will contain information indicating the cause of the failure in DFSMShsm.

DSNU1618I *csect-name* **- RESTORE SYSTEM UTILITY FAILED BECAUSE THERE ARE NO BACKUP SYSTEM HISTORY ENTRIES IN THE BSDS**

Explanation: There are no BACKUP SYSTEM history entries in the BSDS. Without these records no Restore can be done. Backups of the target system(s) must be taken before any RESTOREs can be done.

Severity: 8 (error)

System action: The utility is terminated.

System programmer response: None.

DSNU1619I *csect-name* **- RESTORE SYSTEM UTILITY FAILED, BECAUSE THERE ARE NO BACKUP SYSTEM HISTORY ENTRIES IN THE BSDS WITH A DATA COMPLETION LRSN/RBA PRIOR TO THE LOG TRUNCATION POINT**

Explanation: BACKUP SYSTEM history entries exist in the BSDS. However, none of them indicate a data Backup completion LRSN/RBA with a value less than the log truncation point. This log point is indicated in the SYSPITR parameter of the conditional restart command, CRESTART CREATE, used to truncate the logs.

Severity: 8 (error)

System action: The utility is terminated.

Problem determination: The logs have been truncated to a point before which there are any backups available. The procedure for BACKUP and RESTORE of DB2 systems is described in the Part 4 (Volume 1) of *DB2 Administration Guide* under 'System point in time recovery'. Please review these procedures.

DSNU1621I *csect-name* **-** *utility-name* **UTILITY FAILED BECAUSE THE SYSTEM IS IN SYSTEM RECOVER-PENDING STATE**

Explanation: When the system is in RECOVER-PENDING state, no utility other than RESTORE is allowed to run. An attempt was made to run a utility while the system was in RECOVER-PENDING state.

Severity: 8 (error)

System action: The utility is terminated.

System programmer response: Review the need to run the utility and, if appropriate, resubmit your job after the system has been restored.

DSNU1623I *csect-name* **- THE TERM UTILITY REQUEST IS REJECTED BECAUSE THE SYSTEM IS IN SYSTEM RECOVER PENDING STATE**

Explanation: The TERM request is rejected. The only utility that is allowed to run while the system is in the system RECOVER pending state is the RESTORE utility.

Severity: 4 (warning)

System action: The TERM request is ignored.

DSNU1624I *csect-name* **-** *system-utility* **SYSTEM UTILITY FAILED BECAUSE ONE OR MORE NON-DORMANT MEMBERS ARE IN A FAILED OR NOT NORMALLY QUIESCED STATE. THOSE MEMBERS ARE:** *member-id*

Explanation: The BACKUP or RESTORE of the members of a data sharing group requires that all members be active or normally quiesced. If any members are found not to be in either of these states, the BACKUP or RESTORE cannot continue.

Severity: 8 (error)

System action: The utility is terminated.

System programmer response: Determine why these systems are in a problem state, and whether they can be moved out of this state. Once all members have been returned to either the active or normally quiesced state, resubmit the job.

DSNU1626I *csect-name* **-** *system-utility* **SYSTEM UTILITY FAILED BECAUSE RESTORE SYSTEM IS ALREADY RUNNING OR ANOTHER TASK IS PROCESSING IN A MANNER THAT PREVENTS THIS TASK FROM SERIALIZING PROPERLY**

Explanation: The attempt to acquire the serialization lock failed. In the case of RESTORE SYSTEM, this can only occur if RESTORE SYSTEM is already running. In the case of BACKUP SYSTEM, this can occur if any of the following events are in process:

- System checkpoint
- Data set extensions
- Writing 32K pages
- Writing close page set control log records (PSCRs)
- Creating data sets (for table spaces, indexes, and so on)
- Deleting data sets (for dropping tables spaces, indexes, and so on)
- Renaming data sets (for online reorganizing of table spaces, indexses, and so forth during the switch phase)

Severity: 8 (error)

System action: The utility is terminated.

System programmer response: If it is the RESTORE SYSTEM utility that is being invoked, the only other utility that can be running is RESTORE SYSTEM. If it is the BACKUP SYSTEM utility that is being invoked, any system activity that provokes any of the processes listed in the Explanation above could be the cause of the failure. If possible, determine which utility is responsible and resubmit this job once that utility completes. You may also resubmit your job at a later, less busy time.

DSNU1627I *csect-name* **- RESTORE SYSTEM PRE-LOG APPLY COMPLETED SUCCESSFULLY, COPY POOL =** *copy-pool-name* **TOKEN = X'***token***' ELAPSED TIME =** *hh:mm:ss***.**

Explanation: Restoration of the data from the indicated copy pool with the indicated token has completed successfully.

Severity: 0 (informational)

System action: Continue with log apply phase.

System programmer response: None.

DSNU1628I *csect-name* **- RESTORE SYSTEM PHASE LOG APPLY COMPLETED, ELAPSED TIME =** *hh:mm:ss*

Explanation: The log apply phase of the RESTORE SYSTEM completed in the amount of time that was indicated.

Severity: 0 (informational)

DSNU1629I *csect-name* **- DB2 PUT ONE OR MORE OBJECTS INTO THE RECOVER-PENDING STATE, REBUILD-PENDING STATE, OR LOGICAL PAGE LIST DURING THE LOG APPLY PHASE.**

Explanation: During the log apply process, objects were put into one of the exception states that is indicated in the message. Before these objects can be used, they must be processed again after the Restore completes.

Severity: 0 (informational)

System action: None.

System programmer response: Process those objects in the RECOVER-PENDING or REBUILD-PENDING states with the REBUILD or RECOVER utilities as appropriate. See the Utility Guide and Reference for further details.

To recover pages from the LPL, use the -START DATABASE command with the SPACENAM option, the RECOVER utility, or the LOAD utility.

Problem determination: Use the DISPLAY DATABASE command to identify which objects are in the exception states indicated in the message.

DSNU1631I *csect-name* **- BACKUP SYSTEM UTILITY FAILED BECAUSE THE DFSMShsm CALL FAILED WITH RC = X'***hsm-rc***'. PLEASE SEE THE HSM ACTIVITY LOG FOR HSM MESSAGES INDICATING THE CAUSE OF THE ERROR.**

Explanation: The call to DFSMShsm to perform the Backup failed with the return code indicated in the message. The job log will contain error messages issued by DFSMShsm that will indicate the cause of the error.

Severity: 8 (error)

System action: The utility is terminated.

System programmer response: Review the output and, in particular, the DFSMShsm messages to determine the cause of the failure.

Problem determination: The DFSMSHSM message appearing in the job log will indicate the cause of the failure.

DSNU1632I *csect-name* **- DISPLAY REQUEST WITH MEMBER SPECIFICATION WAS MADE BUT THE SYSTEM IS IN SYSTEM RECOVER-PENDING STATE. THE MEMBER SPECIFICATION IS IGNORED AND ONLY THOSE UTILITIES RUNNING ON THE SYSTEM FROM WHICH THE DISPLAY COMMAND WAS ISSUED WILL BE DISPLAYED**

Explanation: When a system is in the system RECOVER-PENDING state it isn't possible to determine which utilties are executing on members in a data sharing group other than the member on which the DISPLAY command is running. The MEMBER specification is ignored and only those utilities executing on the system on which the command is running are displayed.

Severity: 0 (informational)

System action: Display any utilities running on the system on which the command is running.

DSNU1633I *csect-name* **- BACKUP SYSTEM UTILITY FAILED BECAUSE UNABLE TO WRITE BSDS RECORD DUE TO VSAM ERROR: RC = X'***return code***', RSN = X'***reason code***'**

Explanation: The attempt to update or insert the BACKUP SYSTEM history BSDS record failed due to an error during VSAM processing. The VSAM return and reason codes are indicated in the message.

Severity: 8 (error)

System action: The utility is terminated.

System programmer response: The return and reason code in the message will identify the VSAM error that occurred when attempting to write out the BSDS record. Correct the problem and resubmit the job.

DSNU1634I *csect-name* **- RESTORE SYSTEM UTILITY FAILED BECAUSE THE SYSTEM IS IN TRACKER MODE AND LOGONLY WAS NOT SPECIFIED**

Explanation: The RESTORE SYSTEM utility cannot be run on a tracker system unless the LOGONLY keyword is specified.

Severity: 8 (error)

System programmer response: Change the RESTORE SYSTEM utility control statements to include the LOGONLY parameter.

DSNU1635I *csect-name* **THE RBA RANGE FOR THE LAST CHECKPOINT ISSUED DURING THE LOGAPPLY PHASE OF THE RESTORE SYSTEM UTILITY IS START_RBA = X'***yyyyyyyyyyyy***' END_RBA = X'***zzzzzzzzzzzz***' FOR MEMBER** *mmmm*

Explanation: This message displays the RBA range of the last checkpoint issued by the RESTORE SYSTEM utility. You can use the RBA range as input to DSN1LOGP to print log records that show which objects were affected by the RESTORE SYSTEM utility.

Severity: 0 (informational)

System programmer response: If you suspect a problem, run the DSN1LOGP stand-alone utility to verify which objects were affected by the RESTORE SYSTEM utility.

DSNU1640I *csect-name* **- Z/OS DFSORT LOAD MODULE NOT LOADED**

Explanation: The required z/OS DFSORT load module was not located in the system search order.

Severity: 8 (error)

System action: Utility processing is terminated.

Operator response: Notify the system programmer.

System programmer response: Verify that you correctly configured your system to allow DB2 to use DFSORT load modules. If DFSORT is not installed as your primary z/OS sort product, refer to the DB2 UDB for z/OS Installation Guide's section, install panel DSNTIPW, for information on how to make the DFSORT SORTLPA and SICELINK libraries available for use by DB2 utilities.

DSNU1702I *csect-name* **- LOB PAGE TYPE MISMATCH. ROWID** *rowid* **VERSION** *version* **PAGE** *page-number* **EXPECTED TYPE** *bit-value* **ACTUAL TYPE** *bit-value*

Explanation: A LOB page with page number *page-number* has an unexpected page type specification. In the expected and actual page types, *bit-value* is a 3-bit value. The expected values can be B'100' for a LOB map page or B'110' for a LOB data page. The LOB page in error is part of the LOB with the ROWID *rowid* and version *version*.

Severity: 4

System action: CHECK LOB utility processing continues. The LOB table space status is set to CHECK pending (CHKP).

User response: Notify the system programmer.

System programmer response: Notify IBM Software Support.

DSNU1703I *csect-name* **- LOB PAGE TYPE MISMATCH. PAGE** *page-number* **EXPECTED TYPE** *bit-value* **ACTUAL TYPE** *bit-value*

Explanation: A LOB page with page number *page-number* has an unexpected page type specification. In the expected and actual page types, *bit-value* is a 3-bit value. The expected values can be B'001' for a LOB high-level space map, or B'011' for a LOB low-level space map.

Severity: 4

System action: CHECK LOB utility processing continues. The LOB table space status is set to CHECK pending (CHKP).

User response: Notify the system programmer.

System programmer response: Notify IBM Software Support.

DSNU1704I *csect-name* **- UNEXPECTED PAGE TYPE. ROWID** *rowid* **VERSION** *version* **PAGE** *page-number* **ACTUAL PAGE TYPE** *bit-value*

Explanation: A LOB page with page number *page-number* has an unexpected page type specification. In the actual page type, *bit-value* is a 3-bit value. At the time of processing, the CHECK LOB utility expected either B'100' for a LOB map page or B'110' for a LOB data page. The LOB page in error is part of the LOB with the ROWID *rowid* and version *version*.

Severity: 4

System action: CHECK LOB utility processing continues. The LOB table space status is set to CHECK pending (CHKP).

User response: Notify the system programmer.

System programmer response: Notify IBM Software Support.

DSNU1706I *csect-name* **- PAGE** *page-number* **MISSING IN LOB MAP. ROWID** *rowid* **VERSION** *version*

Explanation: A low-level space map, indicates that the LOB page with page number *page-number* is allocated to the LOB with the ROWID *rowid* and version *version*. The LOB map for the specified LOB does not include the specified page.

Severity: 4

System action: CHECK LOB utility processing continues. The LOB table space status is set to CHECK pending (CHKP).

User response: Notify the system programmer.

System programmer response: Notify IBM Software Support.

DSNU1708I *csect-name* **- LOB MAP PAGE** *page-number* **MISSING. ROWID** *rowid* **VERSION** *version*

Explanation: A low-level space map indicates that the LOB map page with page number *page-number* is allocated to the LOB with the ROWID *rowid* and version *version*. The CHECK LOB utility was unable to verify the existence of this page.

Severity: 4

System action: CHECK LOB utility processing continues. The LOB table space status is set to CHECK pending (CHKP).

User response: Notify the system programmer.

System programmer response: Notify IBM Software Support.

DSNU1710I *csect-name* **- LOB PAGE ALLOCATED TO A DIFFERENT LOB. PAGE** *page-number* **IS IN THE LOB MAP OF ROWID** *rowid1* **VERSION** *version1***. LOW-LEVEL SPACE MAP INDICATES THE PAGE BELONGS TO ROWID** *rowid2* **VERSION** *version2*

Explanation: The LOB page with page number *page-number*, is in the LOB map of the LOB with ROWID *rowid1* and version *version1*. A low-level space map indicates that the page belongs to different LOB with ROWID *rowid2* and version *version2*. The page can be a data page or a LOB map page.

Severity: 4

System action: CHECK LOB utility processing continues. The LOB table space status is set to CHECK pending (CHKP).

User response: Notify the system programmer.

System programmer response: Notify IBM Software Support.

DSNU1712I *csect-name* **- LOB DATA LENGTH INCORRECT. ROWID** *rowid* **VERSION** *version* **EXPECTED LENGTH** *hex-value1* **ACTUAL LENGTH** *hex-value2*

Explanation: The actual sum of the data lengths in the LOB data pages, *hex-value2*, does not match the expected value specified in the LOB map page, *hex-value1*. The LOB in error is identified by ROWID *rowid* and version *version*.

Severity: 4

System action: CHECK LOB utility processing continues. The LOB table space status is set to CHECK pending (CHKP).

User response: Notify the system programmer.

System programmer response: Notify IBM Software Support.

DSNU1714I *csect-name* **- LOB MAP PAGE COUNT INCORRECT. ROWID** *rowid* **VERSION** *version* **EXPECTED COUNT** *hex-value1* **ACTUAL COUNT** *hex-value2*

Explanation: The actual count of data pages, *hex-value1*, does not match the expected number of pages, *hex-value2*, that is specified in the LOB map page. The LOB in error is identified by ROWID *rowid* and version *version*.

Severity: 4

System action: CHECK LOB utility processing continues. The LOB table space status is set to CHECK pending (CHKP).

User response: Notify the system programmer.

System programmer response: Notify IBM Software Support.

DSNU1716I *csect-name* **- NEWER VERSION EXISTS FOR LOB. ROWID** *rowid* **VERSION** *version*

Explanation: The LOB table space contains a newer LOB version for the LOB that is identified by ROWID *rowid* and version *version*. The newer LOB has the same ROWID, *rowid*, but a higher VERSION number. The LOB that is identified by the message is reported as an orphan LOB by the CHECK DATA utility. Other than occupying space in the LOB table space, the presence of orphan LOBs does not hinder LOB processing.

Severity: 0

System action: CHECK LOB utility processing continues. Utility returns completion code=0.

User response: The LOB can be deleted using REPAIR

LOCATE ROWID VERSION DELETE.

DSNU1718I *csect-name* **- LOB PAGE RBA ERROR. ROWID** *rowid* **VERSION** *version* **PAGE** *page-number* **MAXIMUM RBA** *hex-value1* **ACTUAL RBA** *hex-value2*

Explanation: The actual RBA from the LOB data page, *hex-value2*, has exceeded the maximum RBA, *hex-value1*. The maximum RBA is the RBA of the LOB map page. The LOB data page and the LOB map page belong to the LOB with the ROWID *rowid* and version *version*.

Severity: 4

System action: CHECK LOB utility processing continues. The LOB table space status is set to CHECK pending (CHKP).

User response: Notify the system programmer.

System programmer response: Notify IBM Software Support.

DSNU1720I *csect-name* **- PAGE** *page-number* **IN LOB MAP IS MISSING A LOW-LEVEL SPACE MAP ENTRY. ROWID** *rowid* **VERSION** *version*

Explanation: The LOB page that is specified by *page-number* is in the LOB map of the LOB with the ROWID *rowid* and version *version*, but is not allocated in any low-level space map entry.

Severity: 4

System action: CHECK LOB utility processing continues. The LOB table space status is set to CHECK pending (CHKP).

User response: Notify the system programmer.

System programmer response: Notify IBM Software Support.

DSNU1722I *csect-name* **- LOB PAGE CHECK BYTE MISMATCH. ROWID** *rowid* **VERSION** *version* **PAGE** *page-number* **EXPECTED BYTE** *hex-value1* **ACTUAL BYTE** *hex-value2*

Explanation: The actual check byte from the specified LOB data page, *hex-value2*, does not match the expected check byte, *hex-value1*, from the LOB map page of LOB with the ROWID *rowid* and version *version*.

Severity: 4

System action: CHECK LOB utility processing continues. The LOB table space status is set to CHECK pending (CHKP).

User response: Notify the system programmer.

System programmer response: Notify IBM Software Support.

DSNU1724I *csect-name* **-** *page-type* **PAGE NUMBER MISMATCH. ROWID** *rowid* **VERSION** *version* **EXPECTED PAGE NUMBER** *expected-number* **ACTUAL PAGE NUMBER** *actual-number*

Explanation: A LOB page had a page number, *actual-number*, other than the expected number, *expected-number*. The page belongs to the LOB with the ROWID *rowid* and version *version*. *page-type* specifies the type of page that was expected: 'LOB MAP' for an expected LOB map, or 'LOB DATA/MAP' for a page that could be a LOB map or a data page.

Severity: 4

System action: CHECK LOB utility processing continues. The LOB table space status is set to CHECK pending (CHKP).

User response: Notify the system programmer.

System programmer response: Notify IBM Software Support.

DSNU1725I *csect-name* **-** *page-type* **PAGE NUMBER MISMATCH. EXPECTED PAGE NUMBER** *expected-number* **ACTUAL PAGE NUMBER** *actual-number*

Explanation: A LOB page had a page number, *actual-number*, other than the expected number, *expected-number*. *page-type* specifies the type of page that is expected: 'HIGH-LEVEL SPACE MAP' or 'LOW-LEVEL SPACE MAP'.

Severity: 4

System action: CHECK LOB utility processing continues. The LOB table space status is set to CHECK pending (CHKP).

User response: Notify the system programmer.

System programmer response: Notify IBM Software Support.

DSNU1726I *csect-name* **- ZERO PAGE ENCOUNTERED.** *page-type* **PAGE** *page-number* **. ROWID** *rowid* **VERSION** *version*

Explanation: Page numbered *page-number* is a zero page. The page belongs to the LOB with the ROWID *rowid* and version *version*. *page-type* specifies the type of page that is expected: 'LOB MAP' for a LOB map, or 'LOB DATA/MAP' for a LOB map or a data page.

Severity: 4

System action: CHECK LOB utility processing continues. The LOB table space status is set to CHECK pending (CHKP).

User response: Notify the system programmer.

System programmer response: Notify IBM Software Support.

DSNU1727I *csect-name* **- ZERO PAGE ENCOUNTERED.** *page-type* **PAGE** *page-number* **.**

Explanation: Page numbered *page-number* is a zero page. While only the page header information was found to be zero, generally, the entire page will be zero. *page-type* specifies the type of page expected: 'HIGH-LEVEL SPACE MAP' or 'LOW-LEVEL SPACE MAP'.

Severity: 4

System action: CHECK LOB utility processing continues. The LOB table space status is set to CHECK pending (CHKP).

User response: Notify the system programmer.

System programmer response: Notify IBM Software Support.

DSNU1728I *csect-name* **-** *page-type* **PAGE FROM LOB MAP IS MISSING. ROWID** *rowid* **VERSION** *version*

Explanation: A page is in the LOB map of the LOB with the ROWID *rowid* and version *version*, but is not present in the LOB table space. *page-type* specifies the type of missing page, 'LOB MAP' for an addition LOB map or 'LOB DATA/MAP' for a page that could be a LOB map or a data page.

Severity: 4

System action: CHECK LOB utility processing continues. The LOB table space status is set to CHECK pending (CHKP).

User response: Notify the system programmer.

System programmer response: Notify IBM Software Support.

DSNU1730I *csect-name* **- LOB MAP PAGE** *page-number* **CONTAINS ROWID** *rowid1* **VERSION** *version1* **BUT LOW-LEVEL SPACE MAP CONTAINS ROWID** *rowid2* **VERSION** *version2*

Explanation: The LOB map page of a LOB contains the ROWID *rowid1* and version *version1* but the low-level space map indicates that the LOB map page contains the ROWID *rowid2* and version *version2*. The ROWID and version must match.

Severity: 4

System action: CHECK LOB utility processing continues. The LOB table space status is set to CHECK pending (CHKP).

User response: Notify the system programmer.

System programmer response: Notify IBM Software Support.

DSNU1732I *csect-name* **- PAGE** *page-number* **IN LOB MAP HAS AN UNALLOCATED OR DEALLOCATED LOW-LEVEL SPACE MAP ENTRY. ROWID** *rowid* **VERSION** *version*

Explanation: The LOB map of the LOB with the ROWID *rowid* and version *version*, indicates that page *page-number* is part of the LOB. The low-level space map entry for the page indicates that the page is unallocated or deallocated.

Severity: 4

System action: CHECK LOB utility processing continues. The LOB table space status is set to CHECK pending (CHKP).

User response: Notify the system programmer.

System programmer response: Notify IBM Software Support.

DSNU1734I *csect-name* **- UNABLE TO READ** *page-type* **PAGE** *page-number***. ERROR CODE** *hex-value1* **REASON CODE** *hex-value2* **ROWID** *rowid* **VERSION** *version*

Explanation: An attempt to read page number *page-number* failed with error code *hex-value1* and reason code of *hex-value2*. The page belongs to the LOB with the ROWID *rowid* and version *version*. *page-type* specifies the type of page that is expected: 'LOB MAP' for an expected LOB map, or 'LOB DATA/MAP' for a page that could be a LOB map or a data page.

Severity: 4

System action: CHECK LOB utility processing continues. The LOB table space status is set to CHECK pending (CHKP).

User response: Notify the system programmer.

System programmer response: Notify IBM Software Support.

DSNU1735I *csect-name* **- UNABLE TO READ** *page-type* **PAGE** *page-number***. ERROR CODE** *hex-value* **REASON CODE** *hex-value*

Explanation: An attempt to read page number *page-number* failed with error code *hex-value1* and reason code of *hex-value2*. *page-type* specifies the type of page that is expected: 'HIGH-LEVEL SPACE MAP' or 'LOW-LEVEL SPACE MAP'.

Severity: 4

System action: CHECK LOB utility processing

continues. The LOB table space status is set to CHECK pending (CHKP).

User response: Notify the system programmer.

System programmer response: Notify IBM Software Support.

DSNU3030I JOB *jobname* CANCEL ACCEPTED, WAITING FOR DB2 THREADS TO TERMINATE

Explanation: A CANCEL command was issued for a job while a DB2 utility was active. The job will terminate after all of its DB2 threads have terminated.

Severity: 0 (informational)

System action: All threads created by the utility will perform whatever rollback and deallocation processing is required. When this is complete the job will terminate.

System programmer response: Rollback processing might take a significant amount of time, especially for a LOAD LOG YES utility. Under extreme conditions a FORCE command can be used to immediately terminate the job, but this is not recommended because it can cause the thread deallocation to abend.

DSNU3330I *csect-name* - The *utility-name* UTILITY IS PART OF A SEPARATELY LICENSED FEATURE WHICH WAS NOT FOUND ON THIS SYSTEM.

Explanation: DB2 could not find the specified utility in the load library concatenation specified in the submitted JCL. The specified utility is part of an optional feature of DB2.

Severity: 8 (error)

System action: Utility processing does not initiate.

User response: Obtain and install one of the optional utility features (JDB881K or JDB881M) that contains the utility you wish to execute.

DSNU3333I LOAD MODULE *module-name* DID NOT LOAD CORRECTLY. THE *utility-name* UTILITY FUNCTION IS RESTRICTED.

Explanation: The identified load module was found in the product library, but could not be loaded into memory correctly.

Severity: 4 (warning)

System action: Utility processing continues, but the identified utility has restricted function. It might only be executed against objects in the DB2 catalog, the DB2 directory, or the DB2 sample program.

User response: Notify the system programmer.

System programmer response: Verify that the identified load module is reusable and can be re-entered. Also make sure that the identified load module is being loaded from an APF authorized library. This error is normally caused by incorrectly copying a load module or its alias. Use only SMP/E APPLY or the IEWL program to create load modules and their aliases. If SMP/E has produced a load module that cannot be re-entered and is not reusable, then notify IBM Software Support.

DSNU3340I UTILITY PERFORMS DYNAMIC ALLOCATION OF SORT DISK SPACE

Explanation: The utilities that invoke a sort will perform predictable sort disk space dynamic allocation.

Severity: 0 (informational)

System action: Processing of the phase continues.

Programmer response: No action is required.

DSNU3341I CANNOT ALLOCATE SORT DATA SETS WITHIN 255 LIMIT

Explanation: When the utility performs dynamic allocation of sort datasets, the total number of datasets per phase must be less than 255 data sets.

Severity: 8 (error)

System action: Utility processing is terminated.

Programmer response: No action is required.

System programmer response: Provide more disk space for sort work datasets.

DSNU3342I NUMBER OF OPTIMAL SORT TASKS = *nnn*, NUMBER OF ACTIVE SORT TASKS = *nnn*

Explanation: The first number indicates the optimal number of sort tasks for the utility execution. The second number show the maximum number of sort tasks possible for the utility execution.

Severity: 0 (informational)

System action: Processing of the phase continues.

Programmer response: No action is required.

DSNU3343I REAL-TIME STATISTICS INFORMATION MISSING FOR *obj-type obj-qual.obj-name partition-information*

Explanation: The utility could not read necessary information from real-time statistics tables for the specified index or table space. Partitioned objects will include the partition number for which information is missing.

Severity: 0 (informational)

System action: The utility continues, but will use

estimates based on RUNSTATS information for this
object.

System programmer response: Make sure that
real-time statistics are enabled on the system. Run the
REORG or REBUILD INDEX utility on the specified
object to set the base values in the real-time statistics
tables.

Chapter 19. DSNV... messages

DSNV029E LOAD MODULE *module-name* DOES NOT HAVE AMODE(31) ATTRIBUTE

Explanation: During DB2 startup, Agent Services establishes the tasking structure for the control and resource manager address spaces. Three load modules, DSNVEUS3, DSNVEUS4, and DSNVRCT, are loaded and attached as the control programs for each subtask created. All three load modules must have been link-edited with the attributes AMODE(31) and RMODE(ANY). At least one of the modules did not have the 31-bit addressing attribute when it was loaded. The 'module-name' in the error message is the alias name of the first load module whose addressing mode was invalid.

System action: DB2 startup is terminated.

System programmer response: Verify that all installation and maintenance activities against these modules were executed using SMP, that the JCLIN for SMP includes the correct AMODE and RMODE control statements, and that the Linkage Editor provided with the Data Facility Product (DFP) for the version of MVS being used was invoked by SMP for the link-edits. Correct the procedure or JCL which caused the error and rerun the necessary SMP jobs.

Operator response: Notify the system programmer of the failure.

Problem determination: A DB2 dump was requested to a SYS1.DUMP data set.

Load module names and aliases are as follows:

Load module	Alias
DSNVEUS3	DSNVEU3I
DSNVEUS4	DSNVEU4I
DSNVRCT	DSNVRCTI

The MVS service aid AMBLIST provides Linkage Editor module attributes in the summary section of the LISTLOAD function output.

DSNV086E DB2 ABNORMAL TERMINATION REASON=*xxxxxxxx*

Explanation: The DB2 subsystem is being abended, because an error that cannot be corrected has occurred. This message, which is not automatically deleted from the operator console, is issued during DB2 abnormal termination. 'REASON' is the subsystem termination reason code. If DB2 abnormal termination is invoked multiple times, the subsystem termination reason code that accompanies this message is the reason associated with the first invocation. For additional information, look up the reason code in *DB2 Codes*.

System action: Abnormal termination processing continues.

Operator response: Notify the system programmer, and restart DB2.

System programmer response: This message is accompanied by one or more dumps. Obtain a copy of SYS1.LOGREC after DB2 completely terminates; also obtain a copy of the SVC dump(s). If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 5.

DSNV400I ARCHIVE LOG WITH MODE(QUIESCE) CURRENTLY ACTIVE

Explanation: An -ARCHIVE LOG MODE(QUIESCE) command is currently active. This message follows the DSNV401I message as part of the -DISPLAY THREAD command report.

System action: This message is issued as information only. The message indicates that the -ARCHIVE LOG MODE(QUIESCE) command has not completed, and consequently, updates against DB2 resources have been temporarily suspended. This may result in active threads being suspended awaiting termination of the quiesce period. DB2 processing otherwise continues normally.

DSNV401I DISPLAY THREAD REPORT FOLLOWS -

Explanation: This message is always generated as the title for the DISPLAY THREAD command report output. It precedes the other messages generated by this command.

System action: Processing continues normally.

DSNV402I ACTIVE THREADS-

Explanation: This message precedes the listing of active threads.

The series of messages DSNV402I through DSNV405I provide the formatted report for the -DISPLAY THREAD TYPE(ACTIVE) when the status of active threads is requested.

The series of messages DSNV444I through DSNV448I

© Copyright IBM Corp. 1983, 2008

augment the formatted report for the -DISPLAY THREAD TYPE(ACTIVE) when LOCATION, LUWID, or DETAIL keywords are used to show distributed activity.

System action: Processing continues normally.

DSNV403I NAME ST A REQ ID AUTHID PLAN ASID TOKEN

Explanation: This message displays the column headings for the list of threads generated by the -DISPLAY THREAD TYPE(ACTIVE) or -DISPLAY THREAD TYPE(INACTIVE) command.

The message number and subsystem recognition character are not included in this message.

System action: Processing continues normally.

DSNV404I *name status act-ind req-ct corr-id auth-id pname asid token*

Explanation: This message provides the status information on an active or inactive thread and is part of the response to the -DISPLAY THREAD command. Information provided is:

name A 1- to 8-character variable representing the 'connection name' used to establish the thread. For distributed database access threads, this field contains the following:

- For threads using system-directed access and for threads using application-directed access from a DB2 Version 3 or later requester, this variable displays the connection name of the thread at the requesting location.
- For threads using application-directed access from a non-DB2 requester or from a DB2 Version 2 Release 3 requester, this variable displays the constant SERVER.
- If the thread has a status of PT, the connection name contains blanks if the thread of the originating task is running on the same DB2. This helps to show that these parallel tasks are related to the originating task above. If the parallel task is running on a different DB2 for the originating task, then the connection name will be supplied and the entry will be followed by DSNV443I.

If the thread has a status of D, the connection name might be either the connection name for the allied address space or the DB2 subsystem name.

status A 1- or 2- character connection status code with the following values:

D The thread is in the process of termination as a result of the termination of the associated allied task. If this thread is also the last (or only) DB2 thread for the address space, the associated allied task is placed in a wait state.

DA The database access thread slot is currently not associated with a remote connection and is available to be assigned to a type 2 inactive thread.

DI The thread is disconnected from an execution unit. There is no TCB associated with the DB2 thread. This state is only valid when 'connection name'=RRSAF.

N The thread is in either IDENTIFY or SIGNON status.

ND The thread is in either IDENTIFY or SIGNON status, and the thread is currently not associated with any TCB.

PT A parallel task thread was established (plan allocated).

QD The thread is queued for termination as a result of the termination of the associated allied task. If this thread is also the last (or only) DB2 thread for the address space, the associated allied task is placed in a wait state.

QT The CREATE THREAD request was queued. The associated allied task is placed in a wait state.

R2 A distributed thread is performing a remote access on behalf of a request from another location. The thread is currently an inactive connection (type 2 inactive thread) and is waiting for an agent to become available to process.

RA The distributed thread is performing a remote access on behalf of a request from another location.

RK A distributed thread is performing remote access on behalf of a request from another location. The thread is performing an operation that invoked Kerberos services. This status is displayed until Kerberos services returns control to DB2.

RN A distributed thread is performing a remote access on behalf of a request from another location. The request actually accesses data at yet another (DB2) location, so the thread was suspended because DB2 must first connect to the partner location (establish DB2 system conversations with the partner).

When another DB2 site is being contacted for the first time using system-directed access, DB2 must establish DB2 system conversations with the partner location. A request is presented to a DB2 DDF service task that establishes a system conversation. The thread is suspended until the request is processed. If the thread remains in this status for an extended period of time, the DB2 service task that establishes the system conversations might be busy processing other requests. The DB2 DISPLAY LOCATION command ('DISPLAY LOCATION(*) DETAIL') shows you conversation activity for this DB2 system conversation (SYSCON-O) service task.

RQ A distributed thread is performing a remote access on behalf of a request from another location. The thread is suspended because the maximum number of active database access threads (as described by the MAX REMOTE ACTIVE value of the DSN6SYSP macro in the DB2 startup parameter, usually DSNZPARM) was reached. Database access agents (DBAAs) are queued until other DBAAs deallocate or go inactive, providing an available slot. The DBAA resumes when a slot becomes available, and the DBAA is next in the queue. Alternatively, you should consider increasing the MAX REMOTE ACTIVE value.

| **RX** The distributed thread is executing an XA transaction on behalf of a request from another location.

SP A thread is executing within a stored procedure. This status is displayed until the stored procedure terminates and returns control to DB2.

SW A thread is waiting for a stored procedure to be scheduled. This status is displayed until the stored procedure begins execution.

T An allied, nondistributed thread was established (plan allocated).

TD An allied thread was established (plan allocated), and the thread is currently not associated with any TCB.

TN An allied thread was distributed to access data at another DB2 location, but it was suspended because DB2 must first connect to the partner location (establish DB2 system conversations with the partner).

When another DB2 site is being contacted for the first time using system-directed access, DB2 must establish DB2 system conversations with the partner location. A request is presented to a DB2 DDF service task that establishes a system conversation. The thread is suspended until the request is processed. If the thread remains in this status for an extended period of time, the DB2 service task that establishes the system conversations might be busy processing other requests. The DB2 DISPLAY LOCATION command ('DISPLAY LOCATION(*) DETAIL') shows you conversation activity for this DB2 system conversation (SYSCON-O) service task.

TR An allied thread was distributed to access data at another location.

act-ind An asterisk if the thread is active within DB2; blank otherwise.

req-ct A wraparound counter to show the number of DB2 requests.

corr-id A 1- to 12-character variable representing the recovery 'correlation-id' associated with the thread.

For distributed database access threads, this field contains the following:

- While the thread is being created:
  - '028.DBAA nn' - Where nn ranges from 01 to 99. This indicates that the DBAA is in its connection processing, which consists of establishing the DBAA thread, and validating and verifying the user's ID. Refer to the "Controlling Access to a DB2 Subsystem" chapter in Part 3 (Volume 1) of *DB2 Administration Guide* for descriptions of the authorization checks that might be performed.

    RACF requests are serialized on one of the DB2 RACF service tasks, which might increase the validation time when multiple DBAAs are being created concurrently.
- After the thread is created:
  - For threads using system-directed access and threads using application-directed access from a DB2 requester, this variable

displays the correlation-id name of the thread at the requesting location.

– For threads using application-directed access from a non-DB2 requester, this variable displays the first twelve characters in the DDM external name (EXTNAM) parameter of the DDM EXCSAT command received as part of the SQL CONNECT.

authorization ID
The authorization ID associated with a signed-on connection. If not signed-on, this field is blank.

pname A 1- to 8-character variable representing the plan name associated with the thread. For distributed database access threads, this field contains the following:

- For threads using system-directed access and threads using application-directed access from a DB2 Version 3 or later requester, this variable displays the plan name being executed at the requesting location.
- For threads using application-directed access from a non-DB2 requester or from a DB2 Version 2 Release 3 requester, this field displays the constant DISTSERV.

For RRSAF connected threads that did not specify a plan name at connect time, the value ?RRSAF is displayed. If a thread was not established, this field is blank.

If you receive a system plan in this field, see the information about system plans in message DSNT376I.

asid A 1- to 4-character hexadecimal number representing the ASID of the home address space.

token A 1- to 6-character number representing the thread token assigned to the thread. For threads with a status value of 'N', this value will be 0.

The message number and subsystem recognition character are not included in this message.

System action: Processing continues normally.

System programmer response: If you have active threads with QD or D status codes, use the information in message DSN3201I to diagnose a possible DB2 problem.

DSNV405I DISPLAY ACTIVE REPORT COMPLETE

Explanation: This message follows messages DSNV402I through DSNV404I augmented by the series of messages DSNV444I through DSNV448I when LOCATION and/or DETAIL keywords are used to show distributed activity. It indicates that the -DISPLAY THREAD TYPE(ACTIVE) report has completed successfully.

The message number and subsystem recognition character are not included in this message.

System action: Processing continues normally.

DSNV406I INDOUBT THREADS-

Explanation: This message precedes the listing of indoubt threads.

The series of messages DSNV406I through DSNV409I, DSNV446I, DSNV449I through DSNV453I, DSNV457I through DSNV464I, DSNV466I, DSNV467I provides theformatted report for the -DISPLAY THREAD TYPE(INDOUBT) command when the status of indoubt threads is requested.

System action: Processing continues normally.

DSNV407I COORDINATOR STATUS RESET URID AUTHID

Explanation: This message displays the column headings for the list of threads generated by the DISPLAY THREAD TYPE (INDOUBT) command, or for the list of Postponed Abort URs generated by the DISPLAY THREAD TYPE(POSTPONED) or DISPLAY THREAD TYPE(*) command.

The message number and the subsystem recognition character are not included in this message.

System action: Processing continues normally.

DSNV408I *coordinator status reset urid authid*

Explanation: This message lists status information on an indoubt thread or for a Postponed Abort UR.

coordinator
The name of the two-phase commit coordinator.

- If the thread is an allied thread, the coordinator name is either the IMS connection name, the CICS connection name, RRS (for Resource Recovery Services attachment facility connected threads) or the location name of the local DB2 (if it is not one of the above). DSNV449I is displayed if the coordinator is IMS or CICS.
- If the thread is a database access thread, the coordinator name takes one of four forms:

- location:luname -- where "location" is the coordinator's relational database name and "luname" is the coordinator's SNA LU name.
- location:port -- where "location" is the coordinator's relational database name and "port" is the coordinator's TCP/IP port number for two-phase commit resynchronization.
- <luname>:luname -- where "luname" is the coordinator's SNA LU name.
- nnn.nnn.nnn.nnn:port -- where "nnn.nnn.nnn.nnn" is the coordinator's IP address and "port" is the coordinator's TCP/IP port number for two-phase commit resynchronization.

- If the thread is distributed, message DSNV467I is included in the display. This message shows the logical unit of work id and the local token that identifies the distributed thread.

status The status of the thread.

A status of ABORTED is not displayed unless DB2 believes a participant using PN protocols might be indoubt. If the status of a thread for a logical unit of work at a participant is INDOUBT, COMMITTED-H, or ABORTED-H, and the coordinator is a DB2 location that has not cold started, then the lack of any information about the thread in a DISPLAY THREAD(INDOUBT) report indicates that the decision at the coordinator was to abort the logical unit of work. In all other cases, a DSNV408I message for the thread is displayed.

- INDOUBT: The thread is indoubt and continues to appear in the display until all indoubt resolution responsibilities are complete. Additional detail lines that might appear in the report are:
  - DSNV450I and DSNV446I: The names of downstream participants who might be indoubt.
  - DSNV452I or DSNV453I: A heuristic action was taken at a downstream participant. DSNV452I or DSNV453I continues to appear in the display until DB2 resolves the indoubt status with the coordinator. If the coordinator decision is the same as the heuristic decision of the partner, then the detail line is no longer displayed. If the coordinator status is COMMITTED and the heuristic decision of the participant is ABORTED, then the message continues to appear in the display until it is deleted with the RESET INDOUBT command. The database administrators at all involved locations need to know that heuristic damage occurred and at which location.
  - DSNV466I: For a database access thread participant, the elapsed time that the thread was indoubt.
- COMMITTED: The thread is committed. This status is displayed when DB2 is the coordinator and has downstream participants that might be indoubt. The thread appears in the display until all downstream participants complete indoubt resolution. Detail lines DSNV450I and DSNV446I list the downstream participants that have pending resolution. Database locks held for the thread were released.

  If a participant resolved the indoubt thread with a heuristic decision, and the decision was ABORT (rollback), detail line DSNV453I is displayed. This detail line contains the name of the participant and the heuristic decision that it made. Since the coordinator committed and the participant aborted, this indicates heuristic damage. This line continues to appear in the display until it is deleted with the RESET INDOUBT command. The database administrators at all involved locations need to know that heuristic damage occurred and at which location.
- COMMITTED-H: The indoubt thread was heuristically resolved with the RECOVER INDOUBT command. The thread continues to appear in the display until the coordinator is informed of the heuristic decision, and all downstream participants, if any, complete indoubt resolution. Detail lines DSNV450I and DSNV446I list the downstream participants that have pending resolution. Additional detail lines that might appear in the report are:
  - DSNV451I is displayed if resolution with the coordinator is not complete. All database locks were released.
  - DSNV461I is displayed if resolution with the coordinator completed and heuristic damage was detected after the coordinator aborted the logical unit of work. This line continues to appear in the display until it is deleted with the RESET INDOUBT command.
- ABORTED-H: The indoubt thread was heuristically resolved with the RECOVER INDOUBT command. The thread continues to appear in the display until the coordinator is informed of the heuristic decision. Additional detail lines that might appear in the report are:

- DSNV451I is displayed if resolution with the coordinator is not yet complete. All database locks were released.
- DSNV461I is displayed if resolution with the coordinator completed and heuristic damage was detected after the coordinator committed the logical unit of work. This line continues to appear in the display until it is deleted with the RESET INDOUBT command.

- ABORTED: This is a coordinator status and DB2 has one or more downstream participants using the Presumed Nothing (PN) protocol that might be indoubt. The thread appears in the display until all downstream PN participants complete indoubt resolution. Detail lines DSNV450I and DSNV446I list the downstream participants that have pending resolution. Database locks held for the thread were released.

  If a participant resolved the indoubt thread with a heuristic decision, and the decision was COMMIT, detail line DSNV452I is displayed. This detail line contains the name of the participant and the heuristic decision that it made. Since the coordinator aborted and the participant committed, this indicates heuristic damage. This line continues to appear in the display until it is deleted with the RESET INDOUBT command. The database administrators at all involved locations need to know that heuristic damage occurred and at which location.
- ABORT-P: This line represents a Postponed Abort UR. Objects for which this UR has backout work pending are inaccessible (Restart Pending) until the abort is completed (for example, by means of the -RECOVER POSTPONED command).
- ABORT-PSTRT: This line represents a Postponed Abort UR that is currently undergoing -RECOVER POSTPONED processing or automatic DB2 backout processing (requested by restarting with system parameter LBACKOUT = AUTO).

Additional detail lines might appear in an indoubt thread report, as follows:

- Cold Start: When a coordinator cold starts, indoubt participants cannot achieve automatic resolution. A DB2 participant includes message DSNV463I in the display, indicating coordinator cold start and requesting heuristic resolution. Message DSNV458I is included in the display if the coordinator cold started. This message contains information that can be used to identify the thread. This information is provided by the coordinator. A DB2 coordinator provides the connection-name used to establish the thread at the DB2 coordinator and the correlation-id used by that connection.

  When a participant cold starts, the coordinator cannot fulfill its indoubt resolution responsibility. A DB2 coordinator displays message DSNV462I indicating participant cold start. The thread continues to be displayed in the report, even after resolution with all other indoubt participants, if any. The RESET INDOUBT command must be used to purge this thread from the report. The reason for this is that the cold starting participant might have been coordinating downstream participants that are now indoubt. The database administrator at these participants might need to know the status of the thread at the coordinator so correct heuristic decisions can be made.
- Protocol Errors: A communications protocol error might occur during sync point processing and during indoubt resolution processing. These protocol errors are permanent and prevent completion of the automatic indoubt resolution process.

  An indoubt display thread report shows these protocol errors, as follows:
  - A DB2 coordinator detecting a sync point protocol error by a participant, such that the coordinator believes that the participant might be indoubt, adds this thread to the indoubt thread display, if the decision is to commit. Message DSNV460I is included in the display indicating the protocol error and the potential need for heuristic action at the participant. The thread continues to be included in the report until the entry is purged with the RESET INDOUBT command. If the participant is actually indoubt, the database administrator at the participant needs to know the status of the thread so that the correct heuristic decision can be made.
  - A DB2 participant detecting a sync point protocol error by a coordinator when the participant is indoubt adds this thread to the indoubt thread display. Message DSNV459I is included in the display, indicating the protocol error and requesting heuristic resolution. The thread continues to be included in the report until the thread is resolved with the RECOVER INDOUBT command. The DBA at the participant might need to contact the coordinator to determine the status of the thread.

– If a DB2 coordinator detects an indoubt resolution protocol error by a participant, message DSNV457I is included in the indoubt report, indicating the error and requesting heuristic action at the participant. The thread continues to be included in the report until the entry is purged with the RESET INDOUBT command. The database administrator at the participant needs to know the status of the thread so that the correct heuristic decision can be made.
– If a DB2 participant detects an indoubt resolution protocol error by a coordinator, and the thread status is INDOUBT, message DSNV464I is included in the indoubt report, indicating the error and requesting heuristic action. The thread continues to be included in the report until the thread is resolved with the RECOVER INDOUBT command.

reset This column contains YES if the RESET INDOUBT command must be used to purge this thread from the indoubt display.

urid The RBA of the DB2 recovery log. This point is the beginning of recovery logging for this thread.

authid The primary authorization-id associated with the thread.

System action: Processing continues.

DSNV409I DISPLAY INDOUBT REPORT COMPLETE

Explanation: This message follows messages DSNV406I through DSNV408I and indicates that the -DISPLAY THREAD TYPE(INDOUBT) report was generated successfully.

The message number and subsystem recognition character are not included in this message.

System action: Processing continues normally.

DSNV410I NO CONNECTION FOUND FOR NAME=*connection-name*

Explanation: The -DISPLAY THREAD TYPE(ACTIVE) or TYPE(INACTIVE) command was unable to find any connection associated with 'connection-name'. This message is generated once for each requested connection name that was not found.

System action: Command processing continues.

DSNV411I NO THREADS FOUND FOR NAME=*connection-name*

Explanation: The -DISPLAY THREAD TYPE(ACTIVE) or TYPE(INACTIVE) command was unable to locate any threads associated with 'connection-name'. This message is generated once for each connection name for which no threads were found.

System action: Command processing continues.

DSNV412I *csect-name* **NO INDOUBT THREADS FOUND FOR NAME=***connection name*

Explanation: In this message, 'csect-name' is the name of the command program CSECT that issued the message. 'NAME' is the connection name specified on the command.

The named command was unable to locate any indoubt threads associated with the 'connection name'. This message is generated once for each 'connection name' for which no threads were found.

System action: Command processing continues.

DSNV413I *csect-name verbname pkwname* **TERMINATED, DEFAULT UNAVAILABLE**

Explanation: This message indicates that the command ’verbname’ abnormally terminated because the command was entered from an MVS console and a required parameter was not supplied with the command.

csect-name
The name of the command program CSECT that issued the message.

verbname
The command verb as entered by the requester. The error depends on which command was entered.
- DISPLAY THREAD - neither the ’connection-name’, LOCATION, nor LUWID was specified.
- RECOVER INDOUBT - neither ’connection-name’ nor LUWID was specified.

pkwname
The command primary keyword as entered by the requester.

System action: Command processing terminates.

Operator response: Reenter the command specifying one of the required parameters.

DSNV414I THREAD *xxx* COMMIT SCHEDULED

Explanation: In this message, 'xxx' is the recovery identifier specified with the -RECOVER INDOUBT command to identify the indoubt thread and is one of the following:

ID= The recovery correlation-id specified with the command.

NID= The recovery network-id specified with the command.

LUWID = Luwid=token indicates the luwid and token that identify the thread specified with the command.

System action: The thread specified on the -RECOVER INDOUBT command is scheduled for COMMIT recovery action.

DSNV415I THREAD *xxx* ABORT SCHEDULED

Explanation: In this message, 'xxx' is the recovery identifier specified with the -RECOVER INDOUBT command to identify the indoubt thread and is one of the following:

ID= The recovery correlation-id specified with the command.

NID= The recovery network-id specified with the command.

LUWID = The luwid=token indicates the luwid and token that identify the thread specified with the command.

The thread specified on the -RECOVER INDOUBT command is scheduled for ABORT recovery action.

System action: Command processing continues.

DSNV416I THREAD NID=*net-node.number* IS INVALID

Explanation: In this message, 'net-node' is the recovery 'network-id' specified with the command.

The -RECOVER INDOUBT command (NID=net-node.number) determined that the input format for the specified network-id is invalid.

System action: Command processing continues.

Operator response: Prior to reentering the command, ensure that the network-id entered is in the correct format as specified on the -RECOVER INDOUBT command.

DSNV417I THREAD *xxx* NOT FOUND

Explanation: In this message, 'xxx' is the recovery identifier specified with the -RECOVER INDOUBT command to identify the indoubt thread and is one of the following:

ID= The recovery correlation-id specified with the command

NID= The recovery network-id specified with the command

LUWID = The luwid or token specified with the command

The -RECOVER INDOUBT command was unable to locate the specified thread to be scheduled for recovery. Either the thread identifier is incorrect, or the thread no longer is in an indoubt state.

System action: Command processing continues.

Operator response: Before reentering the command, ensure that the thread still is in an indoubt state.

DSNV418I RECOVER INDOUBT REJECTED FOR ID= *correlation-id*

Explanation: In this message, 'ID=' indicates the recovery 'correlation-id' specified with the command.

The -RECOVER INDOUBT command determined that more than one 'network-id' is associated with the specified 'correlation-id'. No recovery action is taken against the indoubt threads associated with the 'correlation-id'.

System action: Command processing continues.

Operator response: To indicate the specific indoubt thread to be scheduled for resolution, the 'network-id' associated with the thread must be specified.

DSNV419I NO CONNECTIONS FOUND

Explanation: The -DISPLAY THREAD command found no active or inactive connections within the system. This message is generated when the command requests thread information for all (TYPE=*), active (TYPE=ACTIVE), or inactive (TYPE=INACTIVE) connections within the system, but no active or inactive connections currently exist.

System action: Command processing continues.

DSNV420I NO INDOUBT THREADS FOUND

Explanation: The -DISPLAY THREAD command found no indoubt threads within the system. This message is generated when the command requests information about all (*) threads currently residing within an indoubt state (TYPE=INDOUBT) within the system, but no threads currently exist within this state.

System action: Command processing continues.

DSNV421I DISPLAY TERMINATED WITH MAX LINES

Explanation: This message indicates the -DISPLAY THREAD TYPE(ACTIVE) or the DISPLAY THREAD TYPE(INACTIVE) report was truncated. The report was generated in response to a command from an MVS console and more than 254 response messages were generated. Only 254 response messages are returned.

The message number and subsystem recognition character are not included in this message.

System action: The truncated report is written to the MVS console.

Operator response: Reissue the -DISPLAY THREAD request specifying a specific connection name, location, luwid, or a combination of these, as appropriate, to further reduce the display.

DSNV422I DISPLAY INDOUBT TERMINATED WITH MAX LINES

Explanation: This message indicates the -DISPLAY THREAD TYPE(INDOUBT) report was truncated. The report was generated in response to a command from an MVS console and more than 254 response messages were generated. Only 254 response messages are returned.

The message number and subsystem recognition character are not included in this message.

System action: The truncated report is written to the MVS console.

Operator response: Reissue the -DISPLAY THREAD request specifying a specific connection name.

DSNV423I DISPLAY THREAD MESSAGE POOL SIZE EXCEEDED

Explanation: The virtual storage requirement needed to generate a -DISPLAY THREAD response display exceeded the maximum size of the message buffer pool.

System action: Processing is terminated.

User response: Reissue the -DISPLAY THREAD request specifying TYPE(INDOUBT), TYPE(ACTIVE), or TYPE(INACTIVE) and a specific connection name, location, luwid, or a combination of these, as appropriate, to further reduce the display.

Operator response: Refer to the User Response.

DSNV424I INACTIVE THREADS-

Explanation: This message precedes the listing of inactive threads.

The series of messages DSNV402I through DSNV405I provide the formatted report for the -DISPLAY THREAD TYPE(INACTIVE) when the status of inactive threads is requested.

The series of messages DSNV444I through DSNV448I augment the formatted report for the -DISPLAY THREAD TYPE(INACTIVE) when LOCATION, LUWID, or DETAIL keywords are used to show idle distributed threads.

System action: Processing continues normally.

DSNV425I DISPLAY INACTIVE REPORT COMPLETE

Explanation: This message follows messages DSNV424I, DSNV403I and DSNV404I, which are augmented by the series of messages DSNV444I through DSNV448I when LOCATION and/or DETAIL keywords are used to show idle distributed threads. It indicates that the -DISPLAY THREAD TYPE(INACTIVE) report completed successfully.

The message number and subsystem recognition character are not included in this message.

System action: Processing continues normally.

DSNV426I *csect-name* **THREAD** *token* **HAS BEEN CANCELED**

Explanation: This message is issued in response to the CANCEL THREAD command. The thread is scheduled for termination.

System action: The specified thread is scheduled for termination. The termination of the thread does not occur unless processing is occurring within DB2.

DSNV427I *csect-name* **INVALID THREAD** *token* **SPECIFIED**

Explanation: This message is issued in response to the CANCEL THREAD command. No thread with the specified *token* was found.

System action: The specified thread is not canceled.

User response: Determine the correct *token* of the thread you want to cancel and reissue the command. You can use the -DISPLAY THREAD command to display all thread activity.

DSNV428I *csect-name* **THREAD** *token* **WAS PREVIOUSLY CANCELED**

Explanation: The thread with the specified token was previously canceled by either the CANCEL THREAD or CANCEL DDF THREAD command, or is already in the process of termination.

The CANCEL command terminates the thread only if the thread is processing in DB2 or is currently suspended by DB2. The thread might not be processing in DB2 in these situations:

- The thread is processing in the application and has not made a request to DB2. The DISPLAY THREAD command indicates this.
- The user placed the thread on an MVS resource using the ENQUE command. The CANCEL command has no effect on the thread until ENQUE completes and the thread returns to DB2.
- The thread is processing in VTAM and is waiting for a response from a partner. The DISPLAY THREAD DETAIL command indicates this. The thread does not terminate until VTAM returns to DB2. VTAM commands, which will terminate the session or LU, can be used to cause VTAM to return to DB2.

DSNV429I V429 CALLING PROCEDURE=*proc*, LOAD MODULE=*loadmod*, PROC=*jclproc*, ASID=*asid*, WLM_ENV=*wlmenv*

Explanation: This message appears in the output of the DISPLAY THREAD command when a thread is running a stored procedure.

proc The name of the stored procedure the SQL application is running.

loadmod The MVS load module associated with the stored procedure. This field will contain blanks until the load module name has been determined.

jclproc The name of the JCL PROC used to start the address space where the stored procedure is running. This field will contain blanks until the stored procedure has been assigned to a specific stored procedure address space.

asid The MVS ASID of the address space where the stored procedure is running. This field will contain 0000 until the stored procedure has been assigned to a specific stored procedure address space.

wlmenv The name of the WLM application environment where the stored procedure is running. If it is running in the DB2-established stored procedure address space, this field will contain blanks.

System action: The DISPLAY THREAD command continues processing.

DSNV431I POSTPONED ABORT THREADS

Explanation: This message precedes the listing of Postponed Abort URs.

The series of messages DSNV431I, DSNV407I, DSNV408I and DSNV432I provide the formatted report for the -DISPLAY THREAD TYPE(POSTPONED) command. The series also provides a section of the formatted report for -DISPLAY THREAD TYPE(*).

DSNV432I DISPLAY POSTPONED ABORT REPORT COMPLETE

Explanation: This message follows messages DSNV431I and DSNV408I and indicates that the -DISPLAY THREAD TYPE(POSTPONED) report was completed successfully, or that the section of the -DISPLAY THREAD TYPE(*) report for Postponed Abort URs has been completed successfully.

System action: Processing continues normally.

DSNV433I NO POSTPONED ABORT THREADS FOUND FOR NAME = *connection name*

Explanation: The -DISPLAY THREAD command found no postponed abort URs for the named connection. This message is generated when the command requests information about threads for the given connection name currently residing within a postponed abort state, but none currently exist within this state.

System action: Command processing continues.

DSNV434I *csect-name* NO POSTPONED ABORT THREADS FOUND

Explanation: This message is issued by the RECOVER POSTPONED command, automatic online backout processing (invoked by restarting with system parameter LBACKOUT=AUTO), and also by the DISPLAY THREAD TYPE(POSTPONED) command. It indicates that no postponed abort threads were found to process. The 'csect-name' is the name of the command program CSECT that issued the message.

System action: The command (or auto-resolution processing) completes normally.

DSNV435I *action* OF POSTPONED ABORT URS HAS BEEN SCHEDULED

Explanation: This message is issued by the RECOVER POSTPONED command, and also by automatic online backout processing (invoked by restarting with system parameter LBACKOUT=AUTO). It indicates that postponed abort URs exist and their backout processing will proceed asynchronously.

ACTION
Description of request

RESOLUTION
Indicates that backout processing for the postponed units of recovery has been requested.

CANCELLATION
Indicates that the CANCEL option was specified on the RECOVER POSTPONED command and that the backout process will be canceled.

System action: The command (or auto-resolution processing) completes normally.

DSNV436I **PGM=***collection.package***, SEC=***section***, STMNT=***statement*

Explanation: This message appears in the output of a DISPLAY THREAD report when a detail display is specified. If DB2 is currently processing an SQL statement, this message identifies the program which contains that statement and displays the SQL statement being executed.

collection
The collection ID associated with *package*. If the SQL statement being executed is associated with a DBRM that was directly bound as part of the plan, and is not associated with a package, then *collection* is not applicable and is indicated by an asterisk (*).

package
The package or DBRM member.

section The SQL section number associated with the package or DBRM.

statement
The SQL statement number associated with the package or DBRM.

The message number and subsystem recognition character are not included in this message. However, V436 is included in the message text to identify the message.

System action: Command processing continues.

DSNV437I **V437-WORKSTATION=***wsname***, USERID=***user-id***, APPLICATION NAME =***appl-name***

Explanation: This message appears in the output of a DISPLAY THREAD report when a non-detail display is specified. If the appropriate information is provided by the client system, this message is issued and identifies more specific information regarding the identity of the end-user associated with this thread.

wsname
The client end-user workstation name.

If not provided by the client system, this field contains an asterisk.

user-id The client end-user user-id.

If not provided by the client system, this field contains an asterisk.

appl-name
The client end-user application name.

If not provided by the client system, this field contains an asterisk.

This message number and subsystem recognition character are not included in this message. However, V437 is included in the message text to identiy the message.

System action: Command processing continues.

DSNV439I *csect-name* **NOBACKOUT OPTION INVALID FOR THREAD** *'token'*

Explanation: This message is issued in response to the CANCEL THREAD command with the NOBACKOUT option. The NOBACKOUT option will not be honored because the canceled thread is part of a global transaction.

System action: The specified thread is not canceled.

DSNV440I **V440-XID=***xid*

Explanation: The thread is managed by an XA transaction manager, such as Websphere, which identifies the transaction with an *xid*. The XID is provided to allow correlation with the XA transaction manager. DB2 uses both the logical unit of work identifier, *luwid*, and the XA transaction identifier, *xid*, to coordinate and recover transactions. DB2 is a participant in an XA transaction that is executing on behalf of an XA transaction manager.

The XID is provided in its hexadecimal representation, and consists of the following elements:

- A 4-byte *Format ID*.
- A 4-byte *Global Transaction Identifier length (gtrid_length)*.
- A 4-byte *Branch Qualifier length (bqual_length)*.
- Variable-length data that is composed of at most two contiguous components: A *Global Transaction Identifier (gtrid)*, and a *Branch Qualifier (bqual)*. The *gtrid_length* element specifies the number of bytes (1-64) that constitute the *gtrid*, starting at the first byte of the data element. The *bqual-length* element specifies the number of bytes (up to 64) that constitute *bqual*, starting at the first byte after *gtrid*.

The message number and subsystem recognition character are not included in this message. However, V440 is included in the message to identify the message.

System action: Command processing continues.

DSNV441I **V441-ACCOUNTING=ACCTNG**

Explanation: This message appears in the output of the DISPLAY THREAD report when a detail display is specified. If the appropriate information is provided by the client system, this message is issued and identifies more specific information regarding the identity of the end-user that is associated with this thread.

acctng The client end-user accounting information. If

the appropriate information is provided by the client system, and if DB2 recognizes the client's format of the information, DB2 will display the user provided portion of the accounting information.

The message number and subsystem recognition character are not included in this message. However, V441 is included in the message text to identify the message.

System action: Command processing continues.

DSNV443I V443-PARALLELISM COORDINATOR=*member-name*, ORIGINATING TOKEN=*token*

Explanation: This message follows a DSNV404I message for each parallel task thread that is running on an assisting DB2. This message gives the DB2 member name and the originating task's thread token.

member-name
The coordinating DB2 member where the originating task thread is running.

token The originating task thread token.

The message number and subsystem recognition character are not included in this message. However, V443 is included in the message text to identify the message.

System action: Command processing continues.

DSNV444I V444-*luwid*=*token* ACCESSING DATA AT.

Explanation: This message follows a DSNV404I message for each thread that was distributed to other locations when a nondetail display is specified. This message gives the logical-unit-of-work identifier for the distributed thread, followed by an equal sign (=) and a token, which can be used in place of luw-id in any DB2 command that accepts luw-id as input. Message DSNV446I follows with the names of the locations associated with this 'luw-id'.

luwid=token
The global logical unit of work id and the local token that identifies the thread. If the thread becomes indoubt, a new token is assigned for the indoubt thread.

The message number and subsystem recognition character are not included in this message. However, V444 is included in the message text to identify the message.

System action: Command processing continues.

DSNV445I V445-*luwid*=*token* ACCESSING DATA FOR *location*

Explanation: This message follows a DSNV404I message for a database access thread.This message displays the logical-unit-of-work identifier assigned to the database access thread.

luwid=token
The global logical unit of work id and the local token that identifies the thread. If the thread becomes indoubt, a new token is assigned for the indoubt thread.

location
Identifies the requesting site. The value specified in *location* is one of the following:

- The relational database name of the requester or the VTAM LU name of the requester (<luname>), a colon (":") delimiter, and the LU name of the requester for SNA connections.
- The dotted decimal IP address of the requester (nnn.nnn.nnn.nnn) for TCP/IP connections.

The message number and subsystem recognition character are not included in this message. However, V445 is included in the message text to identify the message.

System action: Command processing continues.

DSNV446I V446--*location-i location-j* ...

Explanation: This message is a continuation of message DSNV444I or message DSNV450I and provides a list of 'location-i'.

location-i
Identifies the name of a remote DBMS at which data is being accessed or where a thread is indoubt.

The *location-i* string includes two or three tokens delimited by a colon (":"):

- The first token is the partner location name.
- The second token is the partner LU name for an SNA connection, or the dotted decimal IP address for a TCP/IP connection (nnn.nnn.nnn.nnn).
- The third token only appears if the second token is an IP address in which case the third token is a port number.

  When the message appears after message DSNV450I, the port number represents a resync port number. Otherwise the message appears after a DSNV444I message in which case it represents an SQL port number.

A 'location-i' identifier is repeated for each DBMS where data is being accessed or where a thread is indoubt.

The message number and subsystem recognition character are not included in this message. However, V446 is included in the message text to identify the message.

System action: Command processing continues.

DSNV447I V447--LOCATION SESSID A ST TIME

Explanation: This message forms the heading for the listing of distributed active or inactive threads in message DSNV448I. Items in this part of the heading are:

LOCATION
Heading for the associated location names

SESSID
Heading for the associated VTAM-defined session identifiers

A Heading for the associated active indicators

ST Heading for the associated status indicators

TIME Heading for the associated time stamps

See message DSNV448I for descriptions of these items.

The message number and subsystem recognition character are not included in this message. However, V447 is included in the message text to identify the message.

System action: Processing continues normally.

DSNV448I V448--*location sessid active status time*

Explanation: This message follows the DSNV447I message for a database access thread or distributed allied thread when a detail display is specified. This message is displayed for each conversation connected to the thread. It lists detailed information about the conversations associated with the thread.

location
If displaying an allied thread, the location is the name of the server; otherwise, it is the name of the requester. The name can be any of the following: the relational database name of the partner, the VTAM LU name of the partner (<luname>), or the dotted decimal IP address of the partner (nnn.nnn.nnn.nnn).

sessid There are two possible values for this field:

- For VTAM partners, this field contains the VTAM-defined session instance identifier of the session on which the conversation is executing. Zeros if the session identifier is not applicable.

  VTAM does not supply the entire session id to DB2. The first two digits of the session id are incorrect. Therefore, the session id as presented in this message is not acceptable in VTAM commands. The VTAM DISPLAY NET,ID=db2-luname,SCOPE=ACT command must be used to obtain the full session ID. The DISPLAY NET command lists all sessions (SID) for the DB2 logical unit. Scan the DISPLAY NET output for the complete SID.
- For TCP/IP connections, this field contains the local DB2 TCP/IP port number, followed by the partner's TCP/IP port number. The two port numbers are separated by a colon (e.g 5001:28191).

active One of the following:

V The conversation is active within VTAM or TCPIP.

W The conversation is suspended in DB2 waiting for VTAM or TCP/IP notification that the function is complete.

Blank, otherwise.

status Status of the conversation.

This information is presented as two characters, each of which has significance.

First Character

A Conversation in allocation.

C Session limits are being negotiated with the partner (CNOS) prior to conversation allocation.

D Conversation in deallocation.

R Receiving

- At the requesting site, this indicates that the conversation is receiving a response from a request.
- At the server site, this indicates that the conversation is receiving or waiting for a request.

S Sending

- At the requesting site, this indicates that the conversation can send requests to the server.
- At the server site, this indicates that the conversation is sending or preparing to send a response.

X Exchanging log name information with the partner before conversation allocation to determine if the partner supports protected conversations. If the thread remains in this status for an extended period of time, the DB2

service task that exchanges log names might be busy processing other requests. The DB2 DISPLAY LOCATION command ('DISPLAY LOCATION(*) DETAIL') shows you conversation activity for this (RESYNC) service task.

Blank, otherwise.

Second Character

1 Unprotected conversation using system-directed access is active.

2 Unprotected conversation using application-directed access is active.

3 Protected conversation using system-directed access is active.

4 Protected conversation using application-directed access is active.

Blank otherwise.

time The time stamp (yydddhhmmssth) of the last message sent or received on the conversation.

The message number and subsystem recognition character are not included in this message. However, V448 is included in the message text to identify the message.

System action: Command processing continues

DSNV449I V449-HAS NID=*nid* AND ID=*correlation-id*

Explanation: This message is a detail line, which might be displayed with a DISPLAY THREAD TYPE (INDOUBT) report.

nid The network-id assigned by IMS or CICS or, if the connection type is RRSAF, the RRS unit of recovery ID (URID)

correlation-id
The correlation-id assigned by IMS or CICS or if the connection type is RRSAF, the correlation-id assigned by the connected application

The message number and subsystem recognition character are not included in this message. However, V449 is included in the message text to identify the message.

System action: Processing continues normally.

DSNV450I V450-HAS PARTICIPANT INDOUBT AT

Explanation: This message is a detail line which might be displayed with a DISPLAY THREAD TYPE (INDOUBT) report. This message is followed by one or more DSNV446I messages that name the 'locations' where indoubt participant threads are located. The 'luwid' of the indoubt thread is displayed in the preceding DSNV408I message.

The message number and subsystem recognition character are not included in this message. However, V450 is included in the message text to identify the message.

System action: The thread remains indoubt and processing continues normally.

Operator response: See message DSNL406I for more information. The indoubt participants displayed can also result from an abnormal termination of this DB2 subsystem.

DSNV451I V451-RESYNC WITH COORDINATOR STILL PENDING

Explanation: This message is a detail line which might be displayed with a DISPLAY THREAD TYPE (INDOUBT) report. When an indoubt thread is manually resolved with the RECOVER INDOUBT command, this message is included in the report if indoubt resolution with the coordinator was not complete. The 'coordinator' and the 'luwid' of the indoubt thread are displayed in the previous DSNV408I message.

The message number and subsystem recognition character are not included in this message. However, V451 is included in the message text to identify the message.

System action: The thread remains indoubt and processing continues normally.

DSNV452I V452-HEURISTIC COMMIT BY PARTICIPANT AT LOCATION *location*

Explanation: This message is a detail line which might be displayed with a DISPLAY THREAD TYPE (INDOUBT) report. A decision was made at the participant to force the indoubt thread to commit without waiting for automatic resynchronization to be performed. The 'luwid' of the thread is displayed in the preceding DSNV408I message.

location
The 'location' where the HEURISTIC COMMIT was performed. The 'location' is either the relational database name (location-name), the logical unit name (<luname>), or the dotted decimal IP address (nnn.nnn.nnn.nnn) of the participant.

The message number and subsystem recognition character are not included in this message. However, V452 is included in the message text to identify the message.

System action: Processing continues normally.

Operator response: See message DSNL402E for more information.

DSNV453I V453-HEURISTIC ABORT BY PARTICIPANT AT LOCATION *location*

Explanation: This message is a detail line which might be displayed with a DISPLAY THREAD TYPE (INDOUBT) report. A decision was made at the participant to force an indoubt thread to abort without waiting for automatic resynchronization to be performed. The 'luwid' of the thread that was forced to abort is displayed in the preceding DSNV408I message.

location
The 'location' where the HEURISTIC ABORT was performed. The 'location' is either the relational database name (location-name), the logical unit name (<luname>), or the dotted decimal IP address (nnn.nnn.nnn.nnn) of the participant.

The message number and subsystem recognition character are not included in this message. However, V453 is included in the message text to identify the message.

System action: Processing continues normally.

Operator response: See message DSNL403E for more information.

DSNV454I LOCATION OR LUWID KEYWORD INVALID WHEN DDF NOT STARTED

Explanation: The -DISPLAY THREAD TYPE(ACTIVE) command was issued with the keyword LOCATION or LUWID and it was found that the distributed data facility (DDF) was not started. The LOCATION or LUWID keyword can be used only if DDF is in operation.

System action: Command processing is complete.

DSNV456I INVALID PARAMETER WITH LUWID KEYWORD *luwid-parameter*

Explanation: The -DISPLAY THREAD TYPE(ACTIVE), -DISPLAY THREAD TYPE(INACTIVE), or the -RECOVER INDOUBT command was issued with the keyword LUWID and its required parameter list. However, one of the parameters was syntactically invalid as a logical-unit-of-work identifier.

System action: Command processing is complete.

Operator response: The syntax of the DISPLAY THREAD and RECOVER INDOUBT commands is described in *DB2 Command Reference*. Reenter the command specifying valid LUWID values.

DSNV457I V457-RESYNC PROTOCOL ERROR WITH PARTICIPANT *location*

Explanation: This message is a detail line which might be displayed with a DISPLAY THREAD TYPE (INDOUBT) report. Manual resolution of an indoubt thread at the participant location might be required. A protocol error was detected during attempted automatic indoubt resolution with the participant. The 'luwid' of the thread that might require manual resolution at the participant is displayed in the preceding DSNV408I message.

location
The 'location' where manual resolution might be needed. The 'location' is either the relational database name (location-name) of the participant or the logical unit name (<luname>) of the participant.

The message number and subsystem recognition character are not included in this message. However, V457 is included in the message text to identify the message.

System action: Processing continues normally.

Operator response: See message DSNL414E for more information.

DSNV458I V458-CONNECTION-NAME *connection-name* **CORRELATION-ID** *correlation-id*

Explanation: This message is a detail line which might be displayed with a DISPLAY THREAD TYPE (INDOUBT) report. It is displayed only if message DSNV463I is also displayed.

connection-name
The connection-name assigned by the coordinator if the coordinator is another DB2 system. The constant SERVER if the coordinator is not a DB2 system.

correlation-id
The correlation-id assigned by the coordinator if the coordinator is another DB2 system. The first 12 characters of the EXTNAME parameter of the EXCSAT command sent by the coordinator if the coordinator is not a DB2 system.

The message number and subsystem recognition character are not included in this message. However, V458 is included in the message text to identify the message.

System action: Processing continues normally.

Operator response: See message DSNV463I for more information.

DSNV459I V459-SYNCPOINT PROTOCOL ERROR WITH COORDINATOR

Explanation: This message is a detail line which might be displayed with a DISPLAY THREAD TYPE (INDOUBT) report. Manual resolution of an indoubt thread at this location is required. A protocol error was detected during sync point processing with the coordinator. The 'coordinator' and the 'luwid' of the thread that requires manual resolution are both displayed in the preceding DSNV408I message.

The message number and subsystem recognition character are not included in this message. However, V459 is included in the message text to identify the message.

System action: Processing continues normally.

Operator response: See message DSNL412I for more information.

DSNV460I V460-SYNCPOINT PROTOCOL ERROR WITH PARTICIPANT *location*

Explanation: This message is a detail line which might be displayed with a DISPLAY THREAD TYPE (INDOUBT) report. Manual resolution of an indoubt thread at the participant location might be required. A protocol error was detected during sync point processing with the participant. The 'luwid' of the thread that might require manual resolution at the participant is displayed in the preceding DSNV408I message.

location
The 'location' where manual resolution might be needed. The 'location' is either the relational database name (location-name) of the participant or the logical unit name (<luname>) of the participant.

The message number and subsystem recognition character are not included in this message. However, V460 is included in the message text to identify the message.

System action: Processing continues normally.

Operator response: See message DSNL413I for more information.

DSNV461I V461-RESYNC WITH COORDINATOR SHOWS HEURISTIC DAMAGE

Explanation: This message is a detail line which might be displayed with a DISPLAY THREAD TYPE (INDOUBT) report. Indoubt resolution with the coordinator completed. Heuristic damage was detected at this location because the heuristic decision made at this location is different than the decision made at the coordinator. The 'coordinator' and the 'luwid' of the thread are displayed in the previous DSNV408I message. The 'status' field in the previous DSNV408I message shows the heuristic decision made at this location. This message continues to appear until it is deleted with a RESET INDOUBT command.

The message number and subsystem recognition character are not included in this message. However, V461 is included in the message text to identify the message.

System action: The thread remains indoubt and processing continues normally.

Operator response: See message DSNL400E and DSNL401E for more information.

DSNV462I V462-COLD START BY PARTICIPANT AT LOCATION *location*

Explanation: This message is a detail line which might be displayed with a DISPLAY THREAD TYPE (INDOUBT) report. There was an indoubt thread at the participant whose 'luwid' is displayed in the preceding DSNV408I message. DB2 has coordinator responsibility for the indoubt thread and the participant informed DB2 that it performed a cold start operation and lost all knowledge of indoubt threads. There might be heuristic damage at the participant. The RESET INDOUBT command must be used to purge this message from the indoubt report. The reason for this is that the cold-starting participant might have been coordinating downstream participants that are now indoubt. The database administrator at these participants might need to know the status of the thread at the coordinator, so that correct heuristic decisions can be made.

location
The 'location' where the cold start was performed. The 'location' is either the relational database name (location-name), the logical unit name (<luname>), or the dotted decimal IP address (nnn.nnn.nnn.nnn) of the participant.

The message number and subsystem recognition character are not included in this message. However, V462 is included in the message text to identify the message.

System action: Processing continues normally.

Operator response: See message DSNL420E for more information.

DSNV463I V463-COORDINATOR HAS COLD STARTED

Explanation: This message is a detail line which might be displayed with a DISPLAY THREAD TYPE (INDOUBT) report. Manual resolution of an indoubt thread at this location is required. The coordinator for a thread that is indoubt at DB2 was cold started and therefore lost all knowledge of indoubt threads, or the local system's knowledge of the coordinator was cold started, for example, by the "RESET INDOUBT FORCE"

command. The 'coordinator' and the 'luwid' of the thread for which manual resolution is required are displayed in the preceding DSNV408I message.

The message number and subsystem recognition character are not included in this message. However, V463 is included in the message text to identify the message.

System action: Processing continues normally.

Operator response: See message DSNL411I for more information.

DSNV464I V464-RESYNC PROTOCOL ERROR WITH COORDINATOR

Explanation: This message is a detail line which might be displayed with a DISPLAY THREAD TYPE (INDOUBT) report. Manual resolution of an indoubt thread at this location is required. A protocol error was detected during attempted automatic indoubt resolution with the coordinator. The 'coordinator' and the 'luwid' of the thread that requires manual resolution are both displayed in the preceding DSNV408I message.

The message number and subsystem recognition character are not included in this message. However, V464 is included in the message text to identify the message.

System action: Processing continues normally.

Operator response: See message DSNL404E for more information.

DSNV465I V465-THREAD HAS REMAINED IN A PREPARED STATE FOR hh:mm:ss

Explanation: This message might be displayed with a DISPLAY THREAD TYPE (ACTIVE) report. The thread was 'prepared' for the specified period of time. This means that DB2 is the participant in the logical unit of work and completed the first phase of commit and is waiting for the commit or abort decision to be transmitted from the coordinator for the specified period of time. The thread is identified in the preceding DSNV404I message. This message does not appear in a report until the thread was in the prepared state for 60 seconds.

hh:mm:ss
The length of time the thread was prepared (expressed in hours, minutes and seconds). The maximum number of hours reported is 65535, which is roughly 7.5 years. After this time period elapses, the count is reset to zero.

The following actions might be taken to resolve this thread by manual means (that is, take heuristic action) if it is necessary to release locks held by this thread.

- Use the -CANCEL DDF THREAD command to force the thread from the prepared state to the indoubt state. When the thread is indoubt, it appears in the display generated by issuing the DISPLAY THREAD command for type indoubt threads.
- Determine the proper decision (commit or abort) by contacting the coordinator 'location' shown in the display of indoubt threads.
- Use the RECOVER INDOUBT command to heuristically commit or abort the thread.

The message number and subsystem recognition character are not included in this message. However, V465 is included in the message text to identify the message.

This message is issued by the following CSECT: DSNVDTM

System action: Processing continues normally.

DSNV466I V466-THREAD HAS BEEN INDOUBT FOR hh:mm:ss

Explanation: This message is a detail line which might be displayed with a DISPLAY THREAD TYPE (INDOUBT) report. The thread entered the indoubt state at the specified time. This means that DB2 is the participant in the logical unit of work and completed the first phase of commit and is waiting for the commit or abort decision to be transmitted from the coordinator. However, DB2 lost communications with the coordinator. The thread is identified in the preceding DSNV408I message.

hh:mm:ss
The length of time the thread was indoubt (expressed in hours, minutes and seconds). The maximum number of hours reported is 65535, which is roughly 7.5 years. After this time period elapses, the count is reset to zero.

The following actions might be taken to resolve this thread by manual means (that is, take heuristic action) if it is necessary to release locks held by this thread.

- Determine the proper decision (commit or abort) by contacting the coordinator 'location' shown in the display of indoubt threads.
- Use the RECOVER INDOUBT command to heuristically commit or abort the thread.

The message number and subsystem recognition character are not included in this message. However, V466 is included in the message text to identify the message.

System action: Processing continues normally.

DSNV467I V467-HAS LUWID *luwid=token*

Explanation: This message is a detail line, which might be displayed with a DISPLAY THREAD TYPE (INDOUBT) report.

luwid=token
The global logical unit of work id and the local token that identifies the thread.

The message number and subsystem recognition character are not included in this message. However, V467 is included in the message text to identify the message.

System action: Processing continues normally.

DSNV471I V471-*luwid=token*

Explanation: This message follows a DSNV404I message and indicates the agent is currently not associated with a remote location. The agent is available to be assigned to process;
- the completion of a queued receive request on a type 2 inactive thread, or
- a request that a new connection be established, after MAX REMOTE ACTIVE has been reached.

This message displays the logical-unit-of-work identifier assigned to the agent.

luwid=token
The global logical unit of work id and the local token that identifies the agent.

The message number and subsystem recognition character are not included in this message. However, V471 is included in the message text to identify the message.

System action: Command processing continues.

DSNV473I ACTIVE THREADS FOUND FOR MEMBER: *memname***-**

Explanation: This message precedes the listing of active threads found on another member of a data sharing group as the result of the SCOPE(GROUP) keyword.

Message DSNV473I and the series of messages DSNV403I through DSNV405I provide the formatted report for the -DISPLAY THREAD TYPE(ACTIVE) SCOPE(GROUP) when the status of active threads is requested.

The series of messages DSNV444I through DSNV448I augment the formatted report for the -DISPLAY THREAD TYPE(ACTIVE) SCOPE(GROUP) when LOCATION, LUWID, or DETAIL keywords are used to show distributed activity.

memname
Member name for which thread information is being displayed.

System action: Processing continues normally.

DSNV474I INDOUBT THREADS FOUND FOR MEMBER: *memname***-**

Explanation: This message precedes the listing of indoubt threads found on member *memname* as the result of the SCOPE(GROUP) keyword.

Messages DSNV474I, DSNV403I, DSNV404I and DSNV409I provide the formatted report for the -DISPLAY THREAD TYPE(INDOUBT) SCOPE(GROUP) command when the status of indoubt threads is requested.

memname
Member name for which thread information is being displayed.

System action: Processing continues normally.

DSNV475I INACTIVE THREADS FOUND FOR MEMBER: *memname***-**

Explanation: This message precedes the listing of inactive threads found on member *memname* as the result of SCOPE(GROUP) keyword.

Messages DSNV475I, DSNV403I, DSNV404I and DSNV425I provide the formatted report for the -DISPLAY THREAD TYPE(INACTIVE) SCOPE(GROUP) when the status of inactive threads is requested.

The series of messages DSNV444I through DSNV448I augment the formatted report for the -DISPLAY THREAD TYPE(INACTIVE) SCOPE(GROUP) when LOCATION, LUWID, or DETAIL keywords are used to show idle distributed threads.

memname
Member name for which thread information is being displayed.

System action: Processing continues normally.

DSNV476I POSTPONED ABORT THREADS FOUND FOR MEMBER: *memname*

Explanation: This message precedes the listing of Postponed Abort units of recovery found on another member of a data sharing group as the result of the SCOPE(GROUP) keyword.

Messages DSNV476I, DSNV403I, DSNV404I and DSNV432I provide the formatted report for the -DISPLAY THREAD TYPE(POSTPONED) SCOPE(GROUP) command.

memname
Member name for which thread information is being displayed.

System action: Processing continues normally.

DSNV478I SCOPE(GROUP) DISPLAY TERMINATED DUE TO IRLM TRUNCATION

Explanation: This message displays as a result of the -DISPLAY THREAD command when the SCOPE(GROUP) keyword is specified and IRLM has truncated the responses gathered from other members. This is a DB2 internal error.

System action: The -DISPLAY THREAD report is terminated and processing abnormally terminates.

Operator response: Notify the system programmer.

System programmer response: Refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: This is an internal error.

DSNV479I SCOPE(GROUP) DATA UNAVAILABLE IRLM RETURN CODE: *retcode* **IRLM REASON DATA:** *reason*

Explanation: This message displays as a result of the -DISPLAY THREAD command when the SCOPE(GROUP) keyword is specified and the IRLM NOTIFY to other members has returned a bad return code.

retcode Return code from IRLM on the Notify request.

reason Reason code diagnostic information from IRLM on the Notify request.

DSNV480I V480-DB2 IS PARTICIPANT FOR RRS URID=*rrs-urid*

Explanation: This message is a detail line that could be displayed with a DISPLAY THREAD TYPE(ACTIVE) report when the Recoverable Resource Services Unit of Recovery ID (RRS URID) keyword is specified on the command. The RRS URID=*rrs-urid* identifies the RRS URID of the RRS Unit of Recovery that this thread is a participant in.

The message number and subsystem recognition character are not included in this message. However, V480 is included in the mesage text to identify the message.

System action: Command processing continues normally.

DSNV481I V481-DB2 IS COORDINATOR FOR RRS URID=*rrs-urid*

Explanation: This message is a detail line that could be displayed with a DISPLAY THREAD TYPE(ACTIVE) report when the Recoverable Resource Services Unit of Recovery ID (RRS URID) keyword is specified on the command. The RRS URID=*rrs-urid* identifies the RRS URID of the RRS Unit of Recovery that this thread is a coordinator for.

This message is a detail line that also could be displayed with a DISPLAY TYPE(INDOUBT) report when the RRSURID keyword is specified on the command. The RRS URID=*rrs-urid* identifies the RRS URID of the RRS Unit of Recovery that this thread is a coordinator for. When this message appears in an INDOUBT thread report, the preceding DSNV408I message tells the status of the DB2 Unit of Recovery that contains the RRS Unit of Recovery. If manual recovery of the RRS Unit of Recovery is required, the outcome of the DB2 Unit of Recovery can determine whether to COMMIT or ROLLBACK the RRS Unit of Recovery.

The message number and subsystem recognition character are not included in this message. However, V481 is included in the message text to identify the message.

System action: Command processing continues normally.

Chapter 20. DSNW... messages

DSNW001I ASYNCHRONOUSLY GATHERED DATA IS BEING FORMATTED IN THIS DUMP

Explanation: The subsystem formatting exit is not using DB2 summary dump records for formatting. The formatted control blocks may not contain the same values as they did at the time of the error.

System action: Dump formatting continues.

System programmer response: If you want DB2 summary dump records to be used, do not specify the 'SUMDUMP=NO' operand on the dump control statement.

DSNW002I SUMMARY DUMP RECORDS ARE BEING FORMATTED IN THIS DUMP

Explanation: The subsystem formatting exit is using DB2 summary dump record information to format its control blocks.

System action: Dump formatting continues.

System programmer response: If you do not want DB2 summary dump records to be used in formatting, specify the 'SUMDUMP=NO' and 'SUBSYS=subsystem name' on the dump control statement. Both operands are required.

DSNW004I ONE OR MORE INVALID OPERANDS WERE FOUND ON THE DSNWDMP CONTROL CARD. SUBSYSTEM FORMATTING TERMINATED

Explanation: An invalid operand was found on the dump control statement. The operand was not 'SG', 'LG', 'SA', 'AA', or 'ALL'.

System action: Control returns to AMDPRDMP or IPCS to terminate the exit.

System programmer response: Correct the dump control statement operand specified by message DSNW007I.

DSNW005I DUMP FORMATTER WORK BUFFER (DSNWDBUF) CANNOT BE LOCATED

Explanation: The subsystem formatting exit could not locate its work buffer.

System action: Control returns to AMDPRDMP or IPCS to terminate the exit.

System programmer response: Increase the region size.

DSNW006I THE ERLY BLOCK CANNOT BE ACCESSED OR IT IS INVALID

Explanation: The subsystem formatting exit could not locate its anchor block.

System action: Control returns to AMDPRDMP or IPCS to terminate the exit.

System programmer response: Specify 'SUBSYS=subsystem name', 'SUMDUMP=NO' on the dump control statement. The subsystem name should have between 1 and 4 characters.

DSNW007I THE FOLLOWING OPERAND SPECIFIED ON THE DSNWDMP CONTROL CARD IS INVALID: xxxx

Explanation: The specified operand was not one of the valid dump control statement operands.

System action: Control returns to AMDPRDMP or IPCS to terminate the exit.

System programmer response: Check the dump control statement operands.

DSNW008I THE SCOM CANNOT BE ACCESSED OR IT IS INVALID

Explanation: An error was encountered while trying to retrieve the SCOM.

System action: Control returns to AMDPRDMP or IPCS to terminate the exit.

System programmer response: If 'SUMDUMP=NO' was specified on the dump control statement, omit this operand, and resubmit the job. Otherwise, specify this operand on the dump control statement, and resubmit the job.

DSNW009I THE ADDRESS SPACE REQUESTED IS NOT AVAILABLE

Explanation: The DB2 control blocks for the address space specified on the 'SA=asid' dump operand could not be located.

System action: Formatting continues if other dump operands were specified.

System programmer response: Check the ASID specified on the 'SA=' dump operand. The ASID must be specified in hexadecimal.

DSNW010I THE TRACE RMFT CANNOT BE ACCESSED OR IT IS INVALID

© Copyright IBM Corp. 1983, 2008

Explanation: The DB2 trace table could not be located.

System action: Formatting of the DB2 trace table is bypassed, and formatting continues of any other requested dump segment.

System programmer response: If 'SUMDUMP=NO' was specified on the dump control statement, omit this operand. Otherwise, specify this operand on the dump control statement. In either case, resubmit the job.

DSNW011I A LARGER REGION SIZE IS REQUIRED FOR THIS JOB

Explanation: The subsystem formatting exit could not obtain a large enough work buffer to process the summary dump records.

System action: Subsystem formatting continues.

System programmer response: Rerun the job, specifying a larger region size in the JCL.

DSNW013I DMPW NOT FOUND IN SUMMARY DUMP

Explanation: The subsystem formatting exit was unable to locate the DMPW control block in the summary record portion of the dump data set. Because the DMPW provides the main anchor block for the dump formatter, processing is terminated.

System action: Control returns to AMDPRDMP or IPCS to terminate the exit.

System programmer response: If this is not a DB2 dump data set, specify 'SUBSYS=xxxx,SUMDUMP=NO' on the dump control statement (where xxxx is the 1- to 4-character subsystem name), so that the subsystem formatting exit does not attempt to use summary dump data.

DSNW014I REQUIRED SUMMARY DUMP RECORDS ARE NOT IN THIS DUMP. WILL ATTEMPT TO FORMAT FROM NON- SUMMARY DUMP

Explanation: Expected data could not be found in the summary dump. This message is issued for information only. Dump formatting continues. However, it will take longer, because a greater volume of data COMMAND PQUIT must be analyzed.

System action: Formatting is attempted using the full dump instead of the summary dump.

System programmer response: Though processing continues, this message is an indication of a problem, and should be pursued as such.

DSNW015I SSCVT NOT LOCATED. CHECK THE SUBSYSTEM NAME ON THE CONTROL CARD

Explanation: In a search through the SSCVT chain, a match of the subsystem name in the SSCVTs and the subsystem name on the control statement was not found.

System action: Formatting for the named subsystem is not done.

Operator response: Check the 'SUBSYS=' operand on the control statement.

System programmer response: Correctly specify the subsystem name on the control statement.

DSNW016I THE RMVT CANNOT BE ACCESSED OR IT IS INVALID

Explanation: The subsystem formatting exit could not locate the RMVT. The RMVT is required for formatting the DB2 trace table and a number of other DB2 control blocks.

System action: Formatting of the DB2 trace table is bypassed, and formatting of other requested dump segments continues.

System programmer response: If 'SUMDUMP=NO' was specified on the dump control statement, omit this operand. Otherwise, try specifying this operand on the dump control statement. In either case, resubmit the job.

DSNW017I MAXIMUM STACK LEVEL EXCEEDED

Explanation: This condition is usually caused by the DB2 control block formatter looping. The stack array is depleted and can no longer accommodate control blocks.

System action: Dump formatting is terminated.

System programmer response: Examine the control block chains to see if they are looping. If those chains are not looping, increase the dimension of the stack array.

DSNW018I SUBSYS= NOT SPECIFIED ON THE CONTROL CARD OR MISSING. REQUIRED IF SUMDUMP=NO SPECIFIED

Explanation: The 'SUMDUMP=NO' option was specified, but either the 'SUBSYS=' operand is missing, or it was incorrectly specified.

System action: Dump formatting is terminated.

Operator response: Run the dump printing job after the error has been corrected.

DSNW020I UNSUCCESSFUL SEARCH FOR THE ERLY CONTROL BLOCK

Explanation: The 'ERLY' control block could not be located in the dump.

System action: Dump formatting is terminated.

Operator response: Check that the 'SUBSYS=' operand was correctly specified, and rerun the job.

System programmer response: This message can mean that the 'ERLY' control block has been overlaid. Examine the full dump to determine whether the subsystem is present. If it is, you may be able to determine why the control block could not be located.

DSNW022I THE RESIDENT TRACE WAS NOT ACTIVE AT THE TIME OF DUMP

Explanation: Trace table formatting has been attempted, but no trace table existed at the time of the dump.

System action: Dump formatting continues with any other control blocks that were to be formatted.

DSNW023I THE TRACE TABLE ENTRY IS OUT OF SEQUENCE OR OVERLAID

Explanation: A trace entry is overlaid by another trace entry of a different timestamp. This message is issued to flag an unrecognized trace entry. This can occur when the oldest entry in the table is the second half of a double entry.

System action: Formatting of the trace table continues.

DSNW024I TRACE TABLE

Explanation: The trace table follows.

System action: Trace table formatting follows.

DSNW025I DUMP ACCESS ERROR ACCESSING THE TRACE TABLE IN THE DUMP

Explanation: A nonzero return code was returned from the storage access routine when accessing the trace table.

System action: Trace table formatting is bypassed.

System programmer response: Check the address and ASID of the trace table.

DSNW026I CONTROL BLOCK SUMMARY (ALL ADDRESS SPACES)

DSNW027I CONTROL BLOCK SUMMARY (SINGLE ADDRESS SPACE)

DSNW028I CONTROL BLOCK SUMMARY (LONG FORM GLOBAL)

DSNW029I CONTROL BLOCK SUMMARY (SHORT FORM GLOBAL)

Explanation: Messages DSNW026I through DSNW029I provide descriptive information regarding the type of formatting being produced.

System action: Dump formatting continues.

DSNW030I DUMP ACCESS ERROR ACCESSING THE CONTROL BLOCK STRUCTURE TABLE IN THE DUMP

Explanation: The control block structure table cannot be accessed from the dump.

System action: Control block formatting is terminated.

Operator response: Notify the system programmer.

System programmer response: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNW032I DUMP ACCESS ERROR ACCESSING ANCHOR CONTROL BLOCK

Explanation: The anchor control block cannot be accessed from the dump.

System action: Control block formatting is terminated.

Operator response: Notify the system programmer.

System programmer response: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNW033I BEGINNING FORMATTING

Explanation: Formatting of DB2 control blocks is beginning.

DSNW034I TRACE TABLE AND GLOBAL BLOCKS ALREADY DUMPED

Explanation: An indicative dump is being requested. The DB2 trace table and the global blocks have already been dumped with first dump (full dump) for this abend dump (SNAP) invocation. These are, therefore, not dumped in the dump for this task.

DSNW035I WARNING – NO TASK RELATED CONTROL BLOCKS FOR THIS TASK

Explanation: The TCB for which the dump is being requested is not IDENTIFIED to DB2. Task-related control blocks are not dumped. The DB2 trace table and global blocks are dumped only if the 'SYSABEND' DD statement is present and only if this is the first of the dump(s) (full dump) for this abend dump (SNAP) invocation.

System action: No DB2 formatting is done for the specified task.

DSNW036I CONTROL BLOCKS FOR TASKS ASSOCIATED WITH THE ABOVE RECOVERY COORDINATOR TASK

Explanation: The formatted blocks following this message are associated with TCBs that have IDENTIFIED to DB2 with the 'recovery coordinator = no' option. These TCBs may not have invoked SNAP, but they are associated with the TCB that did.

System action: Related control blocks are dumped.

System programmer response: Examine the control blocks for relevant information.

DSNW037I TASK RELATED CONTROL BLOCKS FOR THIS TASK

Explanation: The formatted blocks following this message are associated with the TCB that is being dumped.

System action: Relevant control blocks are dumped.

System programmer response: Examine the control blocks for relevant information.

DSNW038I END OF FORMATTING

Explanation: Formatting of DB2 control blocks is completed.

DSNW039I SUBSYSTEM FORMATTING COMPLETE FOR THIS DUMP

Explanation: The subsystem formatting routine has completed its processing for this dump data set.

DSNW040I 'VVV'CONTROL BLOCK FORMATTER ERROR FLAGS

- X'80': FIELD EXPECTED TO BE ZERO IS NONZERO
- X'40': LAST BLOCK IN CHAIN DOES NOT POINT BACK TO PARENT BLOCK
- X'20': BLOCK NOT FOUND IN DUMP
- X'10': BLOCK LENGTH INCORRECT
- X'08': VERIFICATION CODE INCORRECT
- X'04': BLOCK CHAINED FROM THIS BLOCK NOT FOUND IN DUMP
- X'02': BLOCK CHAINED FROM THIS BLOCK HAS WRONG VERIFICATION CODE
- X'01': VALIDATION CONSTANT DOES NOT MATCH EXPECTED VALUE

Explanation: The message DSNW040I appears at the beginning of a formatted dump. It defines the hexadecimal values associated with the 'VVV=XX' keyword. This keyword is provided on the title line of each control block selected for formatting. It indicates any unusual conditions encountered during the validation and format processing of the current control block structure.

System action: Dump formatting continues.

System programmer response: All nonzero 'XX' values should be examined. If an error is indicated, the printout of the dump should be used for the problem documentation.

Problem determination: Possible errors include:
- X'80'
- X'40'
- X'20'
- X'04'

The current state of the individual control block structures at the time of error might vary with the type of error involved. Examine control block structures with one of these 'XX' values based on the specific ABEND code.

Example: A X'04' associated with the referencing control block and the X'20' associated with the referenced control block may mean the ABEND occurred between the release of the referenced control block and re-initialization of the address within the referencing one.

DB2 validation errors include:
- X'10'
- X'08'
- X'02'
- X'01'

These values indicate that a validation error occurred while formatting the associated control block.

For example, an X'08' associated with the referenced control block indicates the storage area has been overlaid or the address in the referencing control block is incorrect. Subsequent examination of the values in this control block is not reliable.

DSNW041I THE TAB CANNOT BE ACCESSED OR IT IS INVALID

Explanation: The DB2 trace table anchor block could not be located.

System action: Formatting of the DB2 trace table is bypassed, and formatting of any other requested dump segment continues.

System programmer response: If 'SUMDUMP=NO' was specified on the dump control statement, omit this operand. Otherwise, specify this operand on the dump control statement. In either case, resubmit the job.

If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 5.

DSNW049I OLDEST SLOT ADDRESS INVALID, FORMATTING TRACE TABLE FROM FIRST ENTRY

Explanation: The next trace table entry location is invalid.

System action: Dump formatting continues.

System programmer response: The printout of the dump should be used for the problem documentation.

Problem determination: The Trace Anchor Block has been initialized incorrectly, or overlaid during execution.

DSNW050I *ssid*,**NO SDWA/LOGREC, ABND=***comp-reason*,**U=***user-id*, **M=***module*,**C=XYR00.***vrm.comp-function*

Explanation: This message provides the default SVC dump title (SDUMP) associated with the SYS1.DUMP data set when an SDWA was unavailable during recovery processing. The variable fields are:

ssid 4-character DB2 subsystem name

comp-reason
Abend completion code, followed by the abend reason code.

user-id Authorization-ID for the individual subsystem user.

module Functional recovery routine responsible for the dump.

vrm.comp-function
Component-ID, as follows:

vrm DB2 version, release number, and modification level.

comp-function
Component-ID function.

System action: Dump processing continues.

System programmer response: Since the SDWA provides important diagnostic information to assist in problem determination, examine the recovery environment at time of error to determine why an SDWA was not provided for this abend.

In a nonrecovery environment, there may be valid reasons for the lack of an SDWA.

Problem determination: In a recovery environment, functional recovery routines (FRRs) are guaranteed an SDWA by Recovery Termination Manager (RTM). Therefore, the recovery routine is most likely an ESTAE recovery routine. The primary reason for an SDWA not being provided to an ESTAE routine is insufficient storage available during recovery processing. Examine the region sizes allocated to the failing function to ensure that sufficient storage is available.

In a nonrecovery environment, no SDWA is provided because only RTM can create one.

DSNW051I ERROR DURING DUMP PROCESSING

Explanation: This message is generated by the 'SDUMP dump data gathering service' recovery routine when an error is encountered during dump processing.

System programmer response: This error is documented in a SYS1.LOGREC record. This message can be issued because of an error in the invocation of SDUMP, or because of an error in SDUMP itself, or during control block examination and access.

System action: Processing of the 'SUMLSTA' user storage areas is terminated, an SVC dump is requested, and control is returned to RTM.

DSNW052I ERROR ENCOUNTERED DURING LOAD OR VALIDATION OF A CONTROL BLOCK STRUCTURE TABLE MODULE

Explanation: The DB2 dump formatting facility cannot be used to format control blocks. An error occurred during the DB2 startup process while attempting to LOAD one of the Control Block Structures Table (CBDS) modules from the DB2 subsystem program library.

System action: Subsystem startup processing continues.

Operator response: If intervention is desired, notify the system programmer.

System programmer response: If intervention is desired, the DB2 subsystem can be terminated by either an MVS CANCEL command or a -STOP DB2 command. If you suspect an error in DB2, refer to Part

2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: The modules must reside in a DB2 program library named on the execution procedure used to start a DB2 subsystem address space.

The named modules prohibit the use of the DB2 dump formatting facility to format SVC dumps that occur during the current execution cycle of the DB2 subsystem. The named modules are not required for DB2 execution.

DSNW053I VRA DIAGNOSTIC INFORMATION REPORT

Explanation: All DB2 diagnostic information in the Variable Recording Area (VRA) at the time of the error is extracted and displayed in this report.

The address of the SDWA, the contents of SDWARA, and SDWARA's hexadecimal offset into SDWA are also displayed. Individual diagnostic entries are identified by a hexadecimal VRA 'key', followed by the length of the subsequent data. Data fields are displayed in hexadecimal and EBCDIC formats. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

System action: Dump format processing continues.

System programmer response: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNW054I NO VRA DATA RECORDED IN SDWA

Explanation: The SDWA obtained from the SYS1.DUMP data set contained no diagnostic information in the VRA.

System action: VRA report generation is bypassed, dump format processing continues.

System programmer response: If an error is indicated, the printout of the dump should be used for the problem documentation. If VRA data is needed, the abending DB2 subcomponent has failed to supply it. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNW055I FORMAT ROUTINE UNABLE TO LOCATE SDWA

Explanation: The MVS Summary Dump Data Access Service Routine (IEAVTFRD) was unable to locate the SDWA in the summary data portion of the SYS1.DUMP data set.

System action: VRA report generation is terminated and dump format processing continues.

System programmer response: Take the appropriate action, as indicated in the next section. If an error is indicated, use the printout of the dump for problem documentation.

Problem determination: The SDWA is placed in the summary data portion of the SYS1.DUMP data set by AMDPRDMP or IPCS. It indicates either CSA or Local Storage Residency.

If the SDWA is not in the dump data set, no error is indicated.

DSNW056I VRA DIAGNOSTIC REPORT COMPLETE

Explanation: The VRA Diagnostic Report is done.

System action: Dump format processing continues.

DSNW060I BEGIN SAVE AREA TRACE

Explanation: This message identifies the start of the DB2 register Save Area Trace report which appears in the formatted section of a DB2 SVC Dump. The save areas for the current failing agent execution block (EB), and all associated Agent EBs, are traced from the point of error and displayed in order of invocation.

The format of this report is as follows:

Save Area Identification

The address and contents of each save area is displayed and identified by the invoking module.

```
Module-name  Service-level-identifier

SA    .... WD1 ....  HSA ....  LSA ....
RET ....  EPA ....  R0  ....
R1  ....  R2  ....  R3  ....
R4  ....  R5  ....  R6  ....
R7  ....  R8  ....  R9  ....
R10 ....  R11 ....  R12 ....
```

Module-name
Identifies the module responsible for obtaining the save area as indicated in the DB2 module entry point list (MEPL) at the time of the error.

If the MEPL does not contain an entry that corresponds to the module entry point address (EPA), this field is set to:

MODULE INFORMATION UNAVAILABLE

Service-level-identifier
The service level information associated with the module name is obtained from the corresponding module MEPL entry, and consists of the compilation-date and the PTF number of the latest maintenance applied.

If no maintenance has been applied for this module, this field defaults to the Function Modification Identifier (FMID).

SA Identifies the address and contents of each

word of the save area and is displayed using the format and register abbreviation conventions defined by MVS SNAP. All save area data references indicate the primary address space at the time of execution.

Execution Environment

Displays the current execution environment at the time of error for the save areas that follow. This information is repeated each time the current Agent execution EB changed during processing.

```
CURRENT EB=address JOBNAME=name
HASID=hasid PASID=pasid
EXECUTION MODE=mode
```

address
Address of the associated Agent EB.

name The 8-character MVS jobname associated with the home address space.

hasid Home address space identifier.

pasid Primary address space identifier.

mode Agent EB MVS execution mode:
TCB Task Control Mode
SRB Supervisory Request Mode

Register Save Area Contents for Agent EBs

The contents of the register save area chains for the current failing Agent EB, and all suspended Agent EBs at the time of error, are displayed using the save area (SA) format.

```
REGISTERS IN (CB-name) AT TIME OF
execution-status
```

CB-name
Identifies the name of the control block from which the registers for the current Agent EB were obtained:
SDWA MVS System Diagnostic Workarea (Failure).
TCB MVS Task Control Block (Suspend).
STSV MVS SRB Status Save Area (Suspend).

execution-status
Indicates the current execution status of the Agent EB at the time of error:
Failure Failing execution EB.
Suspend
Suspended execution EB.

Change of Primary Address Space

Since all register data references refer to the primary address space associated with the currently active execution EB, the following information is displayed each time the primary address changed during execution:

```
CHANGE OF ADDRESS SPACE, NEW PASID=pasid
```

pasid Indicates the new primary address space identifier associated with the save areas that follow.

System action: save area trace format processing continues for the failing agent EB, and all associated agent EBs.

System programmer response: See the Problem Determination section.

Problem determination: This report may be used to determine the register contents, module invocation sequence, and execution environments leading up to the point of error.

DSNW061I SAVE AREA TRACE COMPLETE

Explanation: This message indicates that the DB2 formatted save area trace report (DSNW060I) is complete.

System action: Dump format processing continues.

DSNW062I R6 (R6-contents) DOES NOT CONTAIN A VALID EB ADDRESS

Explanation: During dump format processing of the DB2 formatted Save Area Trace report (DSNW060I), register 6 (R6) did not contain the address of a valid Agent execution block (EB).

R6-contents:
The contents of register 6.

System action: save area trace format processing is terminated for the current Agent EB, and all prior EBs.

System programmer response: The dump formatter expects the contents of register 6 to contain the address of a valid EB.

Problem determination: Register 6 does not contain the current EB address or a prior EB address.

Refer to the abend reason and completion codes associated with the original error to determine the use of register 6 prior to the failure.

DSNW063I name (address) ASID (asid) NOT FOUND IN DUMP

Explanation: During dump formatter processing of the save area trace report (DSNW060I), a control block or save area was not found in the dump data set.

Since the dump formatter utilizes the DB2 and MVS control blocks defined under the 'name' field of this message to locate individual register save areas, subsequent save areas located via the 'named' control block or save area will not be displayed in the report.

name Identifies the name of the control block or save area that was not found in the dump data set:
SA Indicates a Save Area
ASCE DB2 Address Space Control Element

EB DB2 Execution Block
TCB MVS Task Control Block
RB MVS Request Block
XSB MVS Extended Status Block
PSA MVS Prefix Save Area
SDWA MVS System Diagnostic Work Area
STSV MVS SRB Status Save Area
STKE MVS Cross Memory Stack Element

address
The address of the 'named' control block or save area.

asid The address space identifier associated with the control block or save area.

System action: Register save area trace format processing for the current save area chains is terminated. Subsequent save area processing will vary depending on the specific control block or save area that was available, and the DB2 Agent execution environments at the time of the error.

System programmer response: Due to the execution structures and environmental restrictions of selected DB2 and MVS control structures, some control blocks and save areas associated with these execution environments will not be included in the dump data set.

Problem determination: During MVS/RTM recovery processing, IFC Dump Services attempts to include all control blocks (both DB2 and MVS), and the pertinent DB2 save areas in the dump data set, regardless of the type of error. Control blocks and save areas associated with the following address spaces at time of error will be included in the dump data set:

- Primary, Secondary, and Home address spaces.
- DB2 System Services and Database Services address spaces.

DSNW108I UNABLE TO AUTOMATICALLY START type TRACE

Explanation: System parameters indicated that a DB2 trace should be started automatically during DB2 subsystem initialization, but the DB2 subsystem was unable to start the trace. A nonzero return code was received from the -START TRACE command.

System action: Subsystem initialization continues.

Operator response: Start the trace via the -START TRACE command after DB2 subsystem initialization is complete.

DSNW109I TRACE INITIALIZATION PARAMETERS UNAVAILABLE, DEFAULTS ASSUMED

Explanation: The trace function was unable to access the trace initialization parameters defined by the DSN6SYSP macro. Default values as defined by that macro are assumed for trace parameters.

System action: Subsystem initialization continues.

Operator response: Determine if the system parameters module DSNZPARM is missing or inaccessible. If not, determine if the CSECT DSN6SYSP is missing. Trace can be started with the -START TRACE command.

DSNW120I A SPECIFIED keyword VALUE IS INVALID FOR type TRACE

Explanation: A trace command has been entered, but a specified keyword value is not valid for the trace type requested.

System action: Processing for the TRACE command is terminated.

Operator response: If a -START TRACE command was entered, specify a valid destination for the trace. Otherwise, issue a -DISPLAY TRACE command to determine what traces are currently active. Refer to *DB2 Command Reference* for detailsabout valid destinations.

Problem determination: Obtain a copy of the system or terminal log to verify command entry.

DSNW121I A SPECIFIED CLASS VALUE IS INVALID FOR type TRACE

Explanation: A trace command has been entered, but a specified class value is not valid for the trace type requested.

System action: Processing for the TRACE command is terminated.

Operator response: If a -START TRACE command was entered, specify a valid class for the trace. Otherwise, issue a -DISPLAY TRACE command to determine what options are currently active. Refer to *DB2 Command Reference* for details about valid classes.

Problem determination: Obtain a copy of the system or terminal log to verify command entry.

DSNW122I THE KEYWORD keyword IS NOT VALID FOR type TRACE

Explanation: A trace command has been entered, but 'keyword' is not valid for the trace type specified.

System action: Processing for the TRACE command is terminated.

Operator response: Either the named keyword must be omitted from the command, or a different type of trace must be specified. Refer to *DB2 Command Reference* for details about valid combinations of keywords and trace types.

Problem determination: Obtain a copy of the system or terminal log to verify command entry.

DSNW123I *csect-name* **TRACE RECORDING HAS BEEN RESUMED ON** *dest*

Explanation: 'dest' destination resumes acceptance of trace data after a failure.

System action: Data recording is resumed.

DSNW124I *csect-name type* **TRACE TERMINATED RC=***code* **RMID=***nn*

Explanation: During processing 'type' trace, processing abended due to an error. A trace type of blank indicates all tracing has stopped. 'RMID', displayed in decimal, identifies the resource manager. 'RC', displayed in hexadecimal, specifies the return/reason/abend code associated with the action.

Further collection of the named trace is stopped. If it is necessary to resume collection of the trace, a -START TRACE command may be issued. However if another failure is experienced, the problem should be resolved before starting the trace collection again.

System action: Processing for the named trace type is stopped. The message is not externalized by the functional recovery routine, but is outputted whenever a IFC event is driven at a later time. A trace type of blank indicates all tracing has stopped.

User response: Investigate the reasons for the failure.

Operator response: If necessary to collect the named trace, issue a -START TRACE command to resume processing.

Problem determination: Obtain a copy of the system or terminal log. If a dump was produced, have it printed. Print a copy of the contents of the SYS1.LOGREC for the time period involved. Forward these items along with a copy of the system log to the system programmer.

DSNW125I MULTIPLE ENTRIES NOT ALLOWED FOR keyword AND keyword

Explanation: The following matrix shows keywords which can have multiple entries and the valid combinations. If the row keyword has multiple values then the column keyword can have multiple (M) entries or a maximum of one (1).

Certain keywords are not allowed with some commands. Refer to *DB2 Command Reference* for additional information.

	Class	Dest	Plan	Auth. ID	RMID	TNO	IFCID	Loca
Class		M	M	M	M	M	M	M
Dest	M		M	M	M	M	M	M
Plan	M	M		1	M	1	M	1
Auth. ID	M	M	1		M	1	M	1
RMID	M	M	M	M		M	M	M
TNO	M	M	1	1	M		M	1
IFCID	M	M	M	M	M	M		M
Loca	M	M	1	1	M	1	M	

System action: Processing for the TRACE command is terminated.

Operator response: Reenter the command omitting the invalid keyword.

Problem determination: Obtain a copy of the system or terminal log to verify command entry.

DSNW126I *type* **TRACE NOT ALLOWED, ACTIVE TRACE TABLE FULL**

Explanation: The 'type' trace cannot be started because the active trace table is has reached the maximum number of active traces allowed.

This message is issued by the following CSECT: DSNWVCM1

System action: Processing for the TRACE command is terminated.

Operator response: Use the -DISPLAY TRACE command to see if an active trace could be stopped. An active trace must be stopped before any other start trace command will be processed.

Problem determination: Obtain a copy of the system or terminal log to verify command entry.

DSNW127I CURRENT TRACE ACTIVITY IS–

Explanation: This message is the title for the -DISPLAY TRACE command output. The DSNW127I message is followed by the DSNW128I message, one or more DSNW129I messages, and the DSNW147I message.

This message is issued by the following CSECT: DSNWVCM1

DSNW128I TNO TYPE CLASS DEST QUAL

Explanation: This message is issued before any -DISPLAY TRACE responses. The message follows message DSNW127I. The title indicates the trace number that is active, the type of trace, the class within type, and the destination specified for the active trace entries. Whether the trace is qualified is also displayed. If QUAL=YES, the trace is qualified by authorization ID, plan, RMID, or location. If QUAL=NO, the trace is not qualified. Qualification information can be requested by specifying DETAIL(2) on -DISPLAY TRACE. Refer to messages DSNW143I, DSNW144I, DSNW145I, and DSNW148I for more information about -DISPLAY TRACE. The message number is not displayed to save space on the message line.

This message is issued by the following CSECT: DSNWVCM1

DSNW129I *tno type class dest qual*

Explanation: This message is issued for -DISPLAY TRACE responses. The message indicates the trace number that is active, the type of trace, the class within type, and the destination specified for the active trace entries. Whether the trace is qualified is also displayed. If QUAL=YES, the trace is qualified by authorization ID, plan, RMID, or location. If QUAL=NO, the trace is not qualified. Qualification information may be requested by specifying DETAIL(2) on -DISPLAY TRACE. Refer to messages DSNW143I, DSNW144I, and DSNW145I for trace qualification information.

The message follows message number DSNW128I or DSNW129I. If more information is available than can be displayed on one line, subsequent lines of information are displayed. The message number is not displayed to save space on the message line.

DSNW130I *type* **TRACE STARTED, ASSIGNED TRACE NUMBER** *tno*

Explanation: 'type' trace has been started and assigned the decimal trace number 'tno'. Multiple messages are possible when the start request specifies multiple plans or multiple authorization ids.

This message is issued by the following CSECT: DSNWVCM1

System action: Processing for the -START TRACE command will continue.

DSNW131I STOP TRACE SUCCESSFUL FOR TRACE NUMBER(S) *tno*,...

Explanation: The trace number(s), 'tno,...', have been stopped. Up to five decimal trace numbers may be listed. If more than five traces have been stopped, then another DSNW131I message is sent.

System action: Processing for the -STOP TRACE command will continue.

DSNW132I MODIFY TRACE SUCCESSFUL FOR TRACE NUMBER(S) tno,...

Explanation: The trace numbers, 'tno,...', have been modified. Up to five numbers may be listed. If more than five traces have been modified, then another DSNW131I message is sent.

System action: Processing for the modify command will continue.

DSNW133I *csect-name* **TRACE DATA LOST,** *dest* **NOT ACCESSIBLE RC=***code*

Explanation: The destination specified stopped accepting trace data during a trace. Some condition external to DB2 caused the data rejection. The reason for failure is defined by the return code (RC). The RC hexadecimal value can be:

- The hexadecimal return code from SMF. Refer to the appropriate MVS publication for the specific value.
- The hexadecimal return code from the GTF request.
  04 GTF trace and/or USR tracing is not active.
- The hexadecimal return code from the OPn request.
  04 No OPn buffer is assigned to an application to collect data.
- The hexadecimal return code from the OPX request.
  04 No OPn buffer is assigned to an application to collect data.
  08 The OPX trace destination buffer is full causing data to be lost.
- The hexadecimal return code from the SRV request.
  10 The serviceability routine is absent.
  xx The serviceability routine return code.

System action: Trace processing continues although data will be lost.

This message will also be issued if a -START TRACE command is entered from the console or from the DB2I commands panel to an unassigned OPn or OPX destination. This is because the OP destination is not activated until an application program claims the buffer. DB2 will begin writing trace records to the buffer when an application program issues a -START TRACE to this same OP buffer destination.

Operator response: Investigate the GTF or SMF facility to determine why data is not being accepted. You may want to issue a -START TRACE command to record the data at another destination. The -DISPLAY TRACE command shows what types of data were recorded at the specified destination. Message DSNW133I with RC=8 is due to exhaustion of storage in an OPx buffer.

Problem determination: Obtain a copy of the system or terminal log to verify command. Refer to *MVS/ESA System Management Facilities (SMF)* for an explanation of the return code value. For instances of message DSNW133I with RC=8, consult Appendix E (Volume 2) of *DB2 Administration Guide*. The OPx buffer to which the records were to be written has been filled, and all subsequent records intended for the buffer will be lost until an IFI READA request is issued to read and clear the contents of the buffer. Suggested actions include increasing the size of the OPx buffer and/or issuing the IFI READA request more frequently so that the OPx buffer is read and cleared before buffer storage is exhausted.

DSNW135I *type* **TRACE ALREADY ACTIVE, TRACE NUMBER** *tno*

Explanation: 'type' trace was already active with trace number 'tno'.

System action: Processing for the -START TRACE command will continue.

DSNW137I SPECIFIED TRACE NOT ACTIVE

Explanation: The specified trace could not be found in the active trace table.

System action: Processing for the -STOP or -DISPLAY TRACE command will continue.

Operator response: You may wish to issue an unqualified -DISPLAY TRACE command to determine all the active trace entries. If no traces are active then an unqualified -DISPLAY TRACE command response message is a DSNW137I message.

DSNW138I IFCID *ifcid-number* IS INVALID

Explanation: The specified IFCID number is outside the range of valid IFCID numbers or is an IFCID number which is not allowed on a trace command as specified in Part 5 (Volume 2) of *DB2 Administration Guide*.

System action: Processing of the trace command is terminated before any trace functions are performed.

Operator response: Investigate the reason for the problem.

Problem determination: Refer to *DB2 Command Reference* for the range of valid IFCID numbers.

DSNW139I AUTHORIZATION FAILURE BECAUSE *authorization-privilege* PRIVILEGE REQUIRED

Explanation: The user of the command does not have the required authorization privilege to execute the command.

System action: Command processing is stopped.

User response: Notify the security administrator.

Operator response: Notify the security administrator.

Problem determination: Obtain a copy of the command input and verify the user's current authorization from the DB2 subsystem catalog.

DSNW140I ALL OPN DESTINATIONS ARE ASSIGNED

Explanation: The request could not be processed because destinations OP1 - OP8 were assigned.

System action: Processing of the trace command is terminated before any trace functions are performed.

User response: Investigate the reason for the failure.

Operator response: Investigate the reason for the problem by doing a display trace to analyze the active traces.

Problem determination: Analyze why all the OPn destinations are active after doing a display trace.

DSNW141I STORAGE NOT AVAILABLE FOR BUFFER SIZE *size* K BYTES

Explanation: The request could not be processed because storage cannot be obtained for the OPn destination storage buffer.

System action: Processing of the trace command is terminated.

User response: Investigate the reason for the failure.

Operator response: Investigate the reason for the problem by doing a display trace to analyze the active traces.

Problem determination: The problem is caused either because one or more prior start traces to an OPn destination used all the storage in the IFC OPn storage pool or because ECSA contiguous storage was not available. The IFC OPn storage pool limit is approximately 4 megabytes. When a trace is stopped the buffer storage is not free. If another start trace is issued to the same destination, the storage will be reused. If the requested storage is greater than the current unassigned buffer, it is free and a larger buffer is obtained. If the requested storage is less than the current unassigned buffer, the buffer is reused. Fragmentation can occur if either application programs or an operator specify numerous start commands for varying size buffers.

DSNW142I SPECIFIED BUFSIZE IS INCOMPATIBLE WITH ASSIGNED BUFFER

Explanation: The request could not be processed because the start trace command specified a BUFSIZE value for an OPn destination which has a buffer already assigned with a different BUFSIZE.

System action: Processing of the trace command is terminated.

User response: Investigate the reason for the failure.

Operator response: Investigate the reason for the problem.

Problem determination: The problem may be caused by one application program attempting to start a trace to a destination owned by another program. This is a site problem and can be solved by having application programs start traces to an OPX destination rather than a specific OPn destination. If only one application program is starting traces, then the program should either specify the same BUFSIZE for all requests or should not specify BUFSIZE.

DSNW143I CURRENT TRACE QUALIFICATIONS ARE -

Explanation: This message is issued before -DISPLAY TRACE responses when DETAIL(2) or DETAIL(*) is specified. Message DSNW143I is followed by the

DSNW144I message, one or more DSNW145I messages, and the DSNW148I message.

DSNW144I TNO AUTHID PLAN RMID LOCATION

Explanation: This message is issued before -DISPLAY TRACE responses when DETAIL(2) is specified. The title indicates the trace number that is active, the authorization ID, the plan, the RMID, and the location specified for the active trace entries. The message follows message DSNW143I. The message number is not displayed to save space on the message line.

DSNW145I *tno authid plan rmid location*

Explanation: This message is issued for -DISPLAY TRACE responses when DETAIL(2) is specified. The message indicates the trace number that is active, the authorization ID, the plan, the RMID, and the location specified for the active trace entries.

The message follows messages DSNW143I or DSNW144I. If more information is available than can be displayed on one line, subsequent lines of information are displayed. The message number is not displayed to save space on the message line.

DSNW146I MULTIPLE ENTRIES NOT ALLOWED FOR *keyword*, *keyword* **AND** *keyword*

Explanation: The following matrix shows keywords that can have multiple entries and the valid combinations. If the row keyword has multiple values, the column keyword can have multiple (M) entries or a maximum of one (1).

	Class	Dest	Plan	Auth. ID	RMID	TNO	Loca
Class		M	M	M	M	M	M
Dest	M		M	M	M	M	M
Plan	M	M		1	M	1	1
Auth. ID	M	M	1		M	1	1
RMID	M	M	M	M		M	M
TNO	M	M	1	1	M		1
Loca	M	M	1	1	M	1	

Certain keywords are not allowed with some commands. Refer to *DB2 Command Reference* for additional information.

System action: Processing for the TRACE command is terminated.

Operator response: Omit the invalid keyword and reenter the command.

Problem determination: Obtain a copy of the system or terminal log to verify command entry.

DSNW147I ***END OF DISPLAY TRACE SUMMARY DATA***

Explanation: This message is produced when all summary information for the -DISPLAY TRACE command has been written. Summary information is written when the DETAIL keyword is specified as '1', '*', or not specified at all. Message DSNW147I follows one or more DSNW129I messages. The message number is not displayed to save space on the message line.

System programmer response: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: Collect the following diagnostic items listed in Appendix A, “Problem determination,” on page 659: 1, 2, 5.

DSNW148I ***END OF DISPLAY TRACE QUALIFICATION DATA***

Explanation: This message is produced when all qualification data for the -DISPLAY TRACE command has been written. Qualification data is written when the DETAIL keyword of the -DISPLAY TRACE command is specified as '2', or '*'. Message DSNW148I follows one or more DSNW145I messages. The message number is not displayed to save space on the message line.

System programmer response: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: Collect the following diagnostic items listed in Appendix A, “Problem determination,” on page 659: 1, 2, 5.

DSNW149I STOP TRACE COMMAND WAS UNSUCCESSFUL BECAUSE MONITOR TRACE CLASS 6 IS ACTIVE FOR QUALIFYING TRACE NUMBER(S) *tno*,...

Explanation: No traces were stopped by the -STOP TRACE command. Either -STOP TRACE(*) or -STOP TRACE(MON) CLASS(*) was issued when monitor trace class 6 was active. An asterisk (*) is not allowed for the TRACE and CLASS keywords when monitor trace class 6 is active. The trace numbers, ’tno,...’, indicate the traces that met any other specified -STOP TRACE qualifying criteria (PLAN, AUTHID, RMID, LOCATION, TNO) and also activated monitor trace class 6, thereby preventing all qualifying traces from being stopped. This special handling of the asterisk on the -STOP TRACE command for monitor trace class 6 reduces the possibility of accidentally stopping monitor trace class 6, which enables and disables DB2 data propagation.

System action: No traces are stopped. Processing of the -STOP TRACE command is terminated.

User response: Reissue the -STOP TRACE command without using an asterisk for the trace type. You can use a asterisk for the CLASS keyword to stop all trace types except the monitor (MON) trace.

Operator response: If you do not want DB2 data propagation capability, you can stop monitor trace class 6 to allow full use of the asterisk. First stop the traces indicated by the trace numbers, 'tno,...', and then reissue the command.

If you want DB2 data propagation capability, do not stop monitor trace class 6. You must specify the trace TYPE. You can stop traces without specifying the trace CLASS as long as the trace TYPE is not (MON). If the trace TYPE is (MON), then you must specify the CLASS keyword explicitly (for example, -STOP TRACE(MON) CLASS(1,2)).

Refer to *DB2 Command Reference* for information on restrictions to the -STOP TRACE command when monitor trace class 6 is active.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 5.

Chapter 21. DSNX... messages

DSNX100I bind-type SQL WARNING USING authorization-id AUTHORITY PLAN=plan-id DBRM=dbrm name STATEMENT=statement number object name IS NOT DEFINED

Explanation: The indicated object is not defined in the DB2 catalog.

bind-type
Type of BIND subcommand that caused the warning: BIND, REBIND

authorization-id
Authorization ID used during the BIND process: for a BIND subcommand, indicates the authorization ID of the plan owner; for a REBIND subcommand, indicates the authorization ID of the owner specified by the subcommand, or the latest owner of the plan. If this is a remote rebind, and the OWNER was not specified in the subcommand, then "REMOTE OWNER" will be output.

plan-id Name of the application plan specified in the BIND subcommand, blank if the application plan is not specified.

dbrm name
Name of the DBRM that contains the reference to the indicated object.

statement number
Statement number of the SQL statement referencing the name of the object.

object name
Name of the object that is not defined in the DB2 catalog.

System action: A valid plan is created if no errors are detected and VALIDATE(RUN) is in effect.

System programmer response: Ensure that the indicated table is created before the statement is executed. Otherwise, execution of this statement results in an error.

This statement is bound dynamically on each execution of the statement. For better performance, REBIND the application program after the indicated object is created.

DSNX101I bind-type SQL WARNING USING authorization-id AUTHORITY PLAN=plan-id DBRM=dbrm name STATEMENT=statement number THE SPECIFIED AUTHORIZATION-ID DOES NOT HAVE THE PRIVILEGE TO PERFORM OPERATION op name ON OBJECT object name

Explanation: The identified authorization ID does not have a privilege required for application plan execution.

bind-type
Type of BIND subcommand that caused the warning: BIND or REBIND

authorization-id
Authorization ID used during the bind process. For a BIND subcommand, it is the authorization ID of the plan owner. For a REBIND subcommand, it is the authorization ID of the owner specified by the subcommand, or the latest owner of the plan. If this is a remote operation, and the OWNER was not specified in the subcommand, then REMOTE OWNER is output.

plan-id Name of the application plan specified in the BIND subcommand. If the application plan is not specified, this field is blank.

dbrm name
Name of the DBRM that contains the unauthorized operation.

statement number
Statement number of the SQL statement containing the unauthorized operation.

System action: A valid plan is created if no errors are detected and VALIDATE(RUN) is in effect.

System programmer response: You must take one of the following actions before the statement can be executed. Otherwise, the execution of the statement results in an error.

- Ensure that the appropriate authority is granted to the invoker of the latest BIND subcommand against this application plan
- Invoke the BIND subcommand again using an authorization ID with authority to perform all the operations in this application plan.

This statement is bound dynamically on each execution of the statement. For better performance, rebind the application program after the appropriate authority is

© Copyright IBM Corp. 1983, 2008

granted to the invoker of the BIND subcommand against this application plan.

DSNX102I bind-type SQL WARNING USING authorization-id AUTHORITY PLAN=plan-id DBRM=dbrm name STATEMENT=statement number THE REQUIRED INDEX HAS NOT BEEN CREATED ON TABLE table name IN PARTITIONED TABLESPACE table space name

Explanation: There was an attempt to reference the table in the table space (both specified in the message) when the required index was not yet created.

bind-type
Type of BIND subcommand that caused the warning: BIND or REBIND

authorization-id
Authorization ID used during the bind process. For a BIND subcommand, it is the authorization ID of the plan owner. For a REBIND subcommand, it is the authorization ID of the owner specified by the subcommand, or the latest owner of the plan. If this is a remote operation, and the OWNER was not specified in the subcommand, then REMOTE OWNER is output.

plan-id Name of the application plan specified in the BIND subcommand. If the application plan is not specified, this field is blank.

dbrm name
Name of the DBRM that contains the reference to the indicated table.

statement number
Statement number of the SQL statement referencing the indicated table.

System action: A valid plan is created if no errors are detected and VALIDATE(RUN) is in effect.

System programmer response: Ensure that the required index on the indicated table in the partitioned table space is created before the statement is executed. Otherwise, execution of this statement results in an error. This statement is bound dynamically on each execution of the statement. For better performance, rebind the application program after the required index is created.

DSNX103I bind-type SQL WARNING USING authorization-id AUTHORITY PLAN=plan-id DBRM=dbrm name STATEMENT=statement number BIND ISOLATION LEVEL RR CONFLICTS WITH TABLESPACE LOCKSIZE PAGE OR LOCKSIZE ROW

Explanation: Because of the possible inconsistency between the isolation level of the application and the LOCKSIZE of the table space, table space locking is used to protect the integrity of the application.

bind-type
Type of BIND subcommand that caused the warning: BIND or REBIND

authorization-id
Authorization ID used during the bind process. For a BIND subcommand, it is the authorization ID of the plan owner. For a REBIND subcommand, it is the authorization ID of the owner specified by the subcommand, or the latest owner of the plan. If this is a remote operation, and the OWNER was not specified in the subcommand, then REMOTE OWNER is output.

plan-id Name of the application plan specified in the BIND subcommand. If the application plan is not specified, this field is blank.

dbrm name
Name of the DBRM that contains the reference to the indicated table.

statement number
Statement number of the SQL statement referencing the indicated table.

System action: A valid plan is created if no errors are detected. Table space locking is used. RR isolation level is preserved.

System programmer response: If you do not want table space locking, use isolation level UR, CS or RS.

DSNX104I bind-type SQL WARNING USING authorization-id AUTHORITY PLAN=plan-id DBRM=dbrm name STATEMENT=statement number THE RESULT OF DECIMAL MULTIPLICATION MAY CAUSE OVERFLOW

Explanation: An arithmetic expression contains a decimal multiplication that might cause an overflow condition when the statement is executed. The problem can be corrected by restructuring the arithmetic expression so that decimal multiplication precedes decimal division or by changing the precision and scale of the operands in the arithmetic expression. Refer to Chapter 2 of *DB2 SQL Reference* for the precision and scale of the decimal multiplication and division results.

bind-type
Type of BIND subcommand that caused the warning: BIND or REBIND

authorization-id
Authorization ID used during the bind process. For a BIND subcommand, it is the authorization ID of the plan owner. For a REBIND subcommand, it is the authorization

ID of the owner specified by the subcommand, or the latest owner of the plan. If this is a remote operation, and the OWNER was not specified in the subcommand, then REMOTE OWNER is output.

plan-id Name of the application plan specified in the BIND subcommand. If the application plan is not specified, this field is blank.

dbrm name
Name of the DBRM that contains the reference to the indicated table.

statement number
Statement number of the SQL statement referencing the indicated table.

System action: A valid plan is created if no errors are detected.

DSNX105I bind-type SQL WARNING USING authorization-id AUTHORITY PLAN=plan-id DBRM=dbrm name STATEMENT=statement number SQLCODE=sqlcode SQLSTATE=sqlstateTOKENS=token list

Explanation: An SQL warning code was generated while processing an SQL statement.

bind-type
The type of subcommand that caused the warning: BIND or REBIND

authorization-id
Indicates the authorization ID used during the bind process. For a BIND subcommand, it is the authorization ID of the plan owner. For a REBIND subcommand, it is the authorization ID of the owner specified by the subcommand, or the latest owner of the plan. If this is a remote operation, and the OWNER was not specified in the subcommand, then REMOTE OWNER is output.

plan-id Name of the application plan specified in the BIND subcommand. If the application plan is not specified, this field is blank.

sqlcode
A product specific code which identifies the particular SQL warning condition. The value and semantics of the code are the same as the SQL return codes documented in *DB2 Codes*.

sqlstate
A common code which identifies the particular SQL warning condition.

dbrm name
Name of the DBRM that contains the problem.

statement number
Statement number of the SQL statement containing the problem.

token list
Sequence of one or more tokens separated by blanks. Refer to *DB2 Codes* for an explanation of tokens.

System action: A valid plan is created if no errors are detected.

System programmer response: Refer to *DB2 Codes*, for the particular SQL return code given in this message. If the reason code is negative, this statement is bound dynamically on each execution of the statement. For better performance, rebind the application program after correcting this statement or after the required object is created or altered.

DSNX106I ATTEMPTED REBIND OF PACKAGE pkg-id RESULTED IN A TOTAL OF number ERRORS

Explanation: Errors have been detected during the remote REBIND process. Only the first REBIND error will produce a specific message. Additional errors will simply be totaled, and the number will be returned to the user.

pkg-id Name of the package specified in the subcommand.

number
Total number of errors in the remote REBIND

System action: The package was not rebound.

System programmer response: See previously output message to determine the initial error that caused the subcommand to be terminated unsuccessfully.

DSNX200I bind-type SQL ERROR USING authorization-id AUTHORITY PLAN=plan-id DBRM=dbrm name STATEMENT=statement number SQLCODE=sqlcode SQLSTATE=sqlstateTOKENS=token list CSECT=csect name RDS CODE=location code

Explanation: An error has been detected while processing an SQL statement.

bind-type BIND | REBIND
indicates type of BIND subcommand that caused the error.

authorization-id
indicates the authorization ID used during the bind process. For a BIND subcommand, indicates the authorization ID of the plan owner. For a REBIND subcommand, indicates the authorization ID of the owner specified by the subcommand, or the latest owner of the plan.

plan-id Name of the application plan specified in the BIND subcommand. Blank if the application plan is not specified.

sqlcode
A product specific code which identifies the particular SQL error detected. The value and semantics of the code are the same as the SQL return codes documented in *DB2 Codes*.

sqlstate
A common code which identifies the particular SQL error detected.

dbrm name
Name of the DBRM that contains the error.

statement-number
Statement number containing the SQL error.

token list
Optional sequence of one or more tokens separated by blanks. These are the tokens that are to be substituted into the text of the error code as described at the beginning of *DB2 Codes*.

csect Name of the CSECT detecting the problem.

rds code
Unique code that identifies the location within the CSECT when the problem is detected.

System action: No application plan is created.

System programmer response: Correct the SQL errors in the application program associated with the indicated DBRM. Precompile the application program, and reinvoke the BIND subcommand.

DSNX201I bind-type SQL ERROR USING authorization-id AUTHORITY PLAN=plan-id DBRM=dbrm name STATEMENT=statement number SQL CODE=sql code TOKENS=token list; bind-type DBRM ERROR USING authorization-id AUTHORITY PLAN=plan-id DBRM dbrm name IS IN AN INCONSISTENT STATE

Explanation: The content of a DBRM has been modified after the precompilation process.

bind-type
Type of BIND subcommand that caused the warning: BIND, REBIND

authorization-id
Authorization ID of the plan owner.

plan-id Name of the application plan specified in the BIND or REBIND subcommand (or blank if the application plan is not specified).

dbrm name
Name of the DBRM that is in error.

System action: No application plan is created.

System programmer response: Precompile the application program associated with the indicated DBRM. Reinvoke the BIND subcommand.

DSNX202I INVALID AMODE ATTRIBUTE FOUND FOR DATE OR TIME INSTALLATION REPLACEABLE EXIT ROUTINE module-name

Explanation: During DB2 startup, the Relational Data System subcomponent loads the Date (DSNXVDTX) and Time (DSNXVTMX) exit routines. These routines must have the link-edited attributes AMODE(31) and RMODE(ANY). The module listed in the message 'module-name' did not have the correct addressing attributes. If the date and time LOCAL installation option is specified, DB2 will be unable to execute the date and time functions required by the exit routine that failed to load.

System action: The load module is deleted. DB2 startup will continue.

Operator response: Notify the system programmer.

System programmer response: When DB2 is stopped, the date and time exit routine must be link-edited again with the correct AMODE and RMODE control statements. For more information, refer to Appendix B (Volume 2) of *DB2 Administration Guide*.

DSNX203I LOAD OF THE DATE OR TIME INSTALLATION REPLACEABLE EXIT ROUTINE module-name FAILED WITH MVS ABEND CODE = abend-code REASON CODE = reason-code DB2 STARTUP WILL CONTINUE

Explanation: During DB2 startup, the Relational Data System subcomponent loads the date (DSNXVDTX) and time (DSNXVTMX) exit routines. During the load of EXIT ROUTINE 'module-name', MVS issued the abend code and reason code that would cause abnormal termination of the task, but the abend was suppressed. If the date and time LOCAL option is specified, DB2 will be unable to execute the date and time functions required by the exit routine that failed to load.

System action: The load failed. DB2 startup will continue.

Operator response: Notify the system programmer.

System programmer response: When DB2 is stopped, the date and time exit routine must be link-edited correctly. For more information, refer to Appendix B (Volume 2) of *DB2 Administration Guide* .

DSNX204I *csect-name* **ERROR LOADING THE DB2 CATALOG, REASON** *reason***: DB2 STARTED ACCESS(***MAINT***)**

Explanation: The internal structure (DBD) for the catalog database (DSNDB06) was either not found or was inconsistent when DB2 was started. See the appropriate reason for a more detailed description of the problem.

The value for *reason* is one of the following reason codes:

0 An error occurred reading the DBD01 table space. Check the Console for messages and SYS1.LOGREC for abends.

1 The expected level of the DBD for DSNDB06 is the previous release, but objects exist in the DBD for a current or later release.

2 The expected level of the DBD for DSNDB06 is the current release, but objects exist in the DBD for a later release.

3 The expected level of the DBD for DSNDB06 is a later release, but objects do not exist in the DBD for a later release.

4 The level of the DBD for DSNDB06 is unknown to this version of DB2.

The *mode* is one of the following:

***** The DB01 tablespace could not be opened. Code/Catalog checks are deferred until it can be opened.

MAINT
The DB01 tablespace could be opened and a problem was detected. Prohibits access to any authorization IDs other than install SYSADM and install SYSOPR.

System action: DB2 was started ACCESS(MAINT).

System programmer response: DB2 was started in ACCESS(*mode*). Do not run any CREATES, ALTERS, DROPS, GRANTs, REVOKEs or BINDs when in this state; it might corrupt your catalog and directory. Determine the problem with the DSNDB01.DBD01 table space. An improper REPAIR or RECOVER of the table space might have been run. Fix the problem while in ACCESS(MAINT) mode and then stop and start DB2. See Section 4 (Volume 1) of *DB2 Administration Guide* for more information on recovering the DSNDB01.DBD01 table space.

Problem determination: If you suspect an error in DB2 or the table space is in an inconsistent state refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2.

DSNX205I *csect-name* **THE DATA DEFINITION CONTROL OBJECT DEFINITIONS AND ATTRIBUTES ON MEMBER** *member-name* **ARE NOT THE SAME.**

Explanation: One or more data definition control object definitions and attributes on member *member-name* and on this member are not the same. The data definition control object definitions and attributes are defined in the initialization parameter module DSNZPARM.

System action: Processing continues.

System programmer response: Use the same data definition control table object definitions and attributes for all members of the DB2 data sharing group. See Part 2 of *DB2 Installation Guide* for information about defining data definition control objects and attributes in DSNZPARM.

DSNX206I **LOAD OF THE ACCESS CONTROL AUTHORIZATION INSTALLATION REPLACEABLE EXIT ROUTINE** *module-name* **FAILED WITH MVS ABEND CODE =** *abend-code* **AND REASON CODE =** *reason-code* **DB2 STARTUP WILL CONTINUE**

Explanation: During DB2 startup, the Relational Data System subcomponent loads the Access Control Authorization (DSNX@XAC) exit routine. During the load of EXIT ROUTINE *module-name*, MVS issued the abend code and reason code that indicates abnormal termination of the task. However, the abend was suppressed. DB2 will not use the Access Control Authorization (DSNX@XAC) exit to perform authorization checking. DB2 authorization checking will be performed.

System action: The load failed. DB2 startup will continue.

Operator response: Notify the system programmer.

System programmer response: After DB2 has stopped, you must properly link-edit the Access Control Authorization exit routine. For more information, refer to Appendix B (Volume 2) of *DB2 Administration Guide*.

DSNX207I **INVALID AMODE ATTRIBUTE FOUND FOR ACCESS CONTROL AUTHORIZATION INSTALLATION REPLACEABLE EXIT ROUTINE** *module-name*

Explanation: During DB2 startup, the Relational Data System subcomponent loads the Access Control Authorization (DSNX@XAC) exit routine. This routine must have the link-edited attributes AMODE(31) and RMODE(ANY). The module listed in the message *module-name* did not have the correct addressing

attributes. DB2 will not use the Access Control Authorization (DSNX@XAC) exit to perform authorization checking. DB2 authorization checking will be performed.

System action: The system deletes the load module. DB2 startup will continue.

Operator response: Notify the system programmer.

System programmer response: After DB2 has stopped, you must properly link-edit the Access Control Authorization exit routine with the correct AMODE and RMODE control statements. For more information, refer to Appendix B (Volume 2) of *DB2 Administration Guide*.

DSNX208E *csect-name* **ATTEMPTING TO START DB2 WITH AN UNSUPPORTED CODE LEVEL. REASON CODE** *reason* **MISSING MAINTENANCE:** *maint* **CURRENT LEVEL** *current-code-level* **VALID LEVEL(S)** *valid-code-levels*

Explanation: The DB2 member that is starting has a code level that is not compatible with the catalog or with some other component of DB2.

reason indicates the problem:

1 DB2 is attempting to start with a level of code that is not valid. The missing maintenance, *maint*, is unknown.

2 The level of code that is required by the catalog, which is indicated by *current-code-level*, is not a valid code level. The missing maintenance, *maint*, is unknown.

3 A preconditioning or toleration APAR has not been applied. The required APAR or level of code is specified by *maint*.

4 The preconditioning or toleration maintenance is on the DB2 subsystem, but maintenance to the catalog that is specified by *maint*, has not been applied.

maint is an APAR number or a token that indicates a level of code. The value of *maint* is UNKNOWN if DB2 cannot determine the missing maintenance level.

current-code-level and *valid-code-levels* are for the use of your IBM service representative.

This message can occur in the following situations:

- The starting DB2 fell back from a later release and does not have the proper *maint* applied.
- The starting DB2 does not have the proper coexistence SPE, specified by *maint*, to start in a data sharing group, which is at a later release level.
- The starting DB2 is at a release in which coexistence with the data sharing group is not supported.

System action: DB2 terminates with reason code 00E7009D.

System programmer response: Verify that:

- Coexistence is supported between the releases you are trying to run.
- The DB2 startup procedure points to the correct set of load libraries.
- The correct SPE has been applied.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 16.

DSNX209E *csect-name* **MEMBER** *member-name* **IS AT A CODE LEVEL WHICH IS NOT COMPATIBLE WITH THIS STARTING OR MIGRATING MEMBER. REQUIRED LEVEL:** *new-code-level* **KNOWN LEVELS:** *known-code-levels*

Explanation: The DB2 member that is starting will update the DB2 catalog, but the code level of that DB2 member is not compatible with the code level of the DB2 member that has already started. The DB2 subsystem at the earlier code level requires the correct SPE. The code level information in this message is for the use of your IBM service representative.

System action: DB2 terminates with reason code 00E7009D after detecting the DB2 members that cannot coexist.

System programmer response: Check the DB2 Program Directory and Preventive Service Planning updates (PSP) for more information about the SPE that is needed.

Verify that:

- Coexistence is supported between the releases you are trying to run.
- The DB2 startup procedure points to the correct set of load libraries.
- The correct SPE has been applied to all active members in the group.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 16, 69.

DSNX210I *csect-name* **- ACCESS CONTROL AUTHORIZATION EXIT ROUTINE (DSNX@XAC) HAS INDICATED THAT IT SHOULD NOT BE CALLED, HAS ABENDED OR HAS RETURNED AN INVALID RETURN CODE DURING** *function-code***. RETURN CODE=***return-code* **REASON CODE=***reason-code***. CUMULATIVE ABENDS DURING EXIT PROCESSING=***abend-count***. EXIT ROUTINE STATUS:** *stopped-running***.**

Explanation: The return code from the access control authorization exit routine indicates that it cannot perform authorization checks and should not be called, the exit routine returned an invalid code, or an abend occurred during exit routine processing. This message is issued during the initialization (DB2 startup) call or during an authorization check call to the exit routine, as identified by the function-code. return-code and reason-code are returned in fields EXPLRC1 and EXPLRC2 of the EXPL parameter list for the exit routine.

If the error occurs during initialization, the exit routine will be stopped, as indicated by *stopped-running*=STOPPED. If the message is issued during an authorization check, the exit routine may continue running. During an authorization check, if the message is issued because of an abend during exit routine processing, *abend_count* increases by one and *stopped-running* indicates RUNNING. On the eleventh abend during an authorization check, since DB2 was started, which is indicated by *abend_count* = 000B, the abend tolerance limit has been exceeded and *stopped-running* indicates STOPPED.

System action: DB2 performs different actions, depending on whether this error occurs during DB2 startup or during an authorization check:

- Initialization - DB2 startup:
  - If the return-code is 12 or an unexpected value, then:
    - If *reason_code* is 16, an abend with code 00E70015 occurs and DB2 terminates.
    - If *reason_code* is not 16, an abend with code 00E70009 occurs, DB2 performs all subsequent authorization checks, and the exit is not called again.
  - If an abend occurs during exit routine processing:
    - If the *reason_code* is 16, DB2 terminates.
    - If the *reason_code* is not 16, DB2 performs all subsequent authorization checks, and the exit is not called again.
- Authorization check:
  - If the return-code is 12 or an unexpected value, then:
    - If the initialization *reason_code* was 16, an abend with code 00E70015 occurs and DB2 terminates.
    - If the initialization *reason_code* was not 16, an abend with code 00E70009 occurs, DB2 performs all subsequent authorization checks, and the exit is not called again.
  - If an abend occurs during exit routine processing, a small number of abends are tolerated before the exit is not called again. That number is currently set to 10. The exit routine will continue running until the abend tolerance limit is exceeded. In addition, the following will occur:
    - If the initialization *reason_code* was 16, DB2 terminates.
    - If the initialization *reason_code* was not 16, DB2 performs all subsequent authorization checks.

Operator response: Notify the system programmer.

System programmer response: Refer to the documentation of the access control authorization exit routine for an explanation of *reason_code*. Correct the situation that caused the reason code.

If this message was issued because of an abend during exit routine processing, then search the system log to determine its cause and correct the situation.

If DB2 terminated, restart DB2. If DB2 did not terminate, you need to restart DB2 to reactivate the exit.

Problem determination: Since the access control authorization exit routine is external to DB2, refer to the documentation of the exit routine for an explanation of *return_code* and *reason_code* and a resolution to the problem.

If this message was issued because of an abend during exit routine processing, then determine the cause of the abend from messages in the system log and address it. Look for system messages that occured at approximately the same time as the exit to find more information about the abend.

DSNX211I LOAD OF THE ICSF LOAD MODULE *module-name* FAILED WITH THE RETURN CODE *return-code* AND REASON CODE *reason-code* DB2 STARTUP CONTINUES.

Explanation: DB2 cannot load the ICSF load module *module-name*. Refer to the z/OS ICSF documentation for explanations of the *return-code* reason code.

System action: DB2 startup continues but the encryption/decryption services are not available.

User response: Ensure that the ICSF SCSFMOD0 load module is in the LINKLIST for the z/OS system.

DSNX230I A RESOURCE IS UNAVAILABLE WHEN TRYING TO BUILD THE TRUSTED CONTEXT CACHE REASON=*reason-code* TYPE OF RESOURCE=*resource-type* RESOURCE NAME=*resource-name*

Explanation: During DB2 startup or at the first attempt to create a connection or at a commit related to the TRUSTED CONTEXT DDL statements, an error is encountered while building the trusted context cache and accessing the catalog.

reason-code
The reason code returned by Data Manager during DB2 catalog access.

resource-type
The resource type returned by Data Manager during DB2 catalog access.

resource-name
The resource name returned by Data Manager during DB2 catalog access.

An attempt to rebuild the trusted context cache will be made at the next DB2 startup, or during commit for the TRUSTED CONTEXT related DDL statements or attempt to establish a connection.

System action: The trusted context cache is not built and subsequent connections are not trusted connections, until the problem is resolved. DB2 startup continues if the console message is issued during the startup. If the message is issued during a remote connect attempt to use the trusted connection, SQLCODE +20360 is returned. If the message is issued when DSN ASUSER is specified, DSNE137I message is returned.

If the message is issued during the commit for TRUSTED CONTEXT related DDL statements, the DB2 catalog is updated, but the trusted context cache is not updated.

Operator response: Notify the system programmer.

System programmer response: The message is issued when an error is encountered while trying to access the catalog tablespace, DSNDB06.SYSCONTX. The reason code indicates the cause of the failure. Verify the identity of the resource that was not available. Check if there is any prior abend messages in the console. If you suspect an error in DB2, refer to Part 3 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSNX900E *csect-name* **ILLEGAL VALUE DETECTED IN** *colname* **COLUMN OF SYSIBM.SYSPROCEDURES FOR PROCEDURE** *procname*

Explanation: The *colname* column of the SYSIBM.SYSPROCEDURES catalog table contains an unsupported value in a row associated with PROCEDURE=*procname*.

colname The name of the column of the SYSIBM.SYSPROCEDURES table that contains the illegal value.

procname
The name of the stored procedure being processed when the illegal value was detected.

System action: The invalid row in the SYSIBM.SYSPROCEDURES table cannot be processed.

Operator response: Contact the system programmer.

System programmer response: Use ALTER PROCEDURE or ALTER FUNCTION to correct the unsupported column value in the SYSIBM.SYSROUTINES table.

DSNX905I *csect-name* **PROCEDURE OR FUNCTION** *name* **TERMINATED ABNORMALLY ASID=** *asid* **WLM_ENV=** *wlmenv*

Explanation: A stored procedure terminated abnormally while attempting to satisfy an SQL CALL statement for an SQL application.

name The name of the stored procedure or function that terminated abnormally.

asid The MVS ASID of the address space where the stored procedure executed.

wlmenv The name of the WLM application environment where the stored procedure executed. If it executed in the DB2-established stored procedures address space, this field will contain blanks.

System action: The SQL CALL statement fails, and the SQL application is forced to roll back.

System programmer response: Correct the cause of the abnormal termination. If the correction requires the stored procedure to be linkedited again, then it might be necessary to cause the corrected MVS load module to be reloaded. Different commands must be used to cause this reloading depending on the environment in which the stored procedure was executed.

If it was run in a WLM-established stored procedures address space, issue the MVS VARY WLM,APPLENV=name,REFRESH command.

If it was run in the DB2-established stored procedures address space, issue the -START PROCEDURE command.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 4, 5.

DSNX906I *csect-name* **PROCEDURE OR FUNCTION** *name* **TERMINATED ABNORMALLY. THE PROCEDURE HAS BEEN STOPPED. ASID=** *asid* **WLM_ENV=** *wlmenv*

Explanation: A stored procedure terminated abnormally while attempting to satisfy an SQL CALL statement for an SQL application.

name The name of the stored procedure or function that terminated abnormally

asid The MVS ASID of the address space where the stored procedure executed.

wlmenv The name of the WLM application environment where the stored procedure

executed. If it executed in the DB2-established stored procedures address space, this field will contain blanks.

System action: The SQL CALL statement fails, and the SQL application is forced to roll back.

The abnormal termination count for procedure *procname* was reached. The procedure is stopped and further requests for the procedure are rejected.

System programmer response: Correct the cause of the abnormal termination. If the correction requires the stored procedure to be linkedited again, then it might be necessary to cause the corrected MVS load module to be reloaded. Different commands must be used to cause this reloading depending on the environment in which the stored procedure was executed.

If it was run in a WLM-established stored procedures address space, issue the MVS VARY WLM,APPLENV=name,REFRESH command.

If it was run in the DB2-established stored procedures address space, issue the -START PROCEDURE command.

After the appropriate command has been issued to refresh the stored procedure load module,use the -START PROCEDURE command to activate the stored procedure.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 4, 5.

DSNX908I *csect-name* **PROCEDURE OR FUNCTION** *name* **WITH LOAD MODULE** *module* **EXCEEDED CPU RESOURCE LIMIT SSN=** *ssn* **PROC=** *jclproc***ASID=** *asid* **WLM_ENV=** *wlmenv*

Explanation: A stored procedure or function consumed more CPU service units than allowed by the ASUTIME column of the ALTER PROCEDURES catalog table. The stored procedure might be in a loop.

name The name of the stored procedure or function being processed

module The MVS load module associated with the stored procedure

ssn The name of the DB2 subsystem that called the stored procedure.

jclproc The name of the JCL PROC used to start this stored procedures address space.

asid The MVS ASID of the address space where the stored procedure executed.

wlmenv The name of the WLM application environment where the stored procedure executed. If it executed in the DB2-established stored procedures address space, this field will contain blanks.

System action: DB2 detaches the TCB associated with the stored procedure, which causes abnormal termination of the stored procedure.

Operator response: Contact the system programmer.

System programmer response: If the stored procedure is looping, correct the stored procedure.

If the stored procedure is performing correctly, increase the ASUTIME value in the CREATE PROCEDURES row associated with the stored procedure.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 4, 5.

DSNX921I *csect-name* **SOME SQL CALL STATEMENTS CANNOT BE PROCESSED, BECAUSE REQUIRED RESOURCES ARE NOT AVAILABLE REASON=***reason* **TYPE=***type* **NAME=***name*

Explanation: DB2 was unable to validate the attributes of the SYSIBM.SYSROUTINES catalog table and related objects because of an unavailable resource.

reason The DB2 reason code describing the unavailable resource

type The resource type

name The name of the resource

System action: SQL CALL statements are rejected.

System programmer response: Resolve the unavailable resource problem. Then use the -START PROCEDURE(*) command to cause DB2 to accept SQL CALL statements.

Operator response: Note the *reason* code, the *type* code, and the *name*. See the Problem Determination section for an explanation of the collected data.

Problem determination: The reason code tells why the resource is unavailable. Reason codes are described in "Part 4. Reason Codes". Use the table defined for message DSNT500I to find the type of resource identified in the message.

DSNX922I *csect-name* **BEGINNING DISCONNECTION OF STORED PROCEDURE ADDRESS SPACES**

Explanation: This message is produced during -STOP DB2 processing. It indicates that DB2 SQL processing has been quiesced, and DB2 is now beginning the termination process for the stored procedures address spaces.

DSNX923I *csect-name* **ALL STORED PROCEDURE ADDRESS SPACES ARE NOW DISCONNECTED.**

Explanation: This message is produced during -STOP DB2 processing. It indicates that all of the stored procedure address spaces are now disconnected from DB2 and the stop of DB2 will continue.

WLM-established stored procedure address spaces are terminated by WLM after the disconnection from DB2.

DSNX940I *csect-name* **- DISPLAY PROCEDURE REPORT FOLLOWS-**

Explanation: This message is the title for the report output from the DISPLAY PROCEDURE command. It precedes messages DSNX941I and DSNX942I. Message DSNX943I might also be displayed. The final message for the report is DSNX945I.

System action: The DISPLAY PROCEDURE command processing continues.

DSNX941I PROCEDURE STATUS ACTIVE QUED MAXQ TIMEOUT FAIL WLM_ENV

Explanation: This message displays the column headings for the list of stored procedures generated by the DISPLAY PROCEDURE command. The definitions are:

PROCEDURE
The name of the stored procedure

STATUS
The status of the stored procedure

ACTIVE
The number of DB2 agents that are currently running this stored procedure

QUED The number of DB2 agents that are currently waiting for this stored procedure to be scheduled

MAXQ The maximum number of DB2 agents that have waited for this stored procedure to be scheduled since DB2 was started

TIMEOUT
The number of times a request for this procedure timed out while waiting to be scheduled since DB2 was started

FAIL The Number of times this procedure has abended since it was last started

WLM_ENV
The MVS Workload Manager environment name for this stored procedure

System action: The DISPLAY PROCEDURE command processing continues.

DSNX942I *procedure status active qued maxq timeout fail wlm_env*

Explanation: This message lists the DISPLAY PROCEDURE information for a given stored procedure. The definitions of the columns are:

procedure
The name of the stored procedure.

status The status of the stored procedure. The status can be any of the following values:

STARTED
The stored procedure is able to accept SQL CALL requests.

STOPABN
The stored procedure was stopped because of the previous abnormal termination of the stored procedure application. SQL CALL requests for the stored procedure are rejected.

The -START PROCEDURE command can be used to place the stored procedure in STARTED status.

STOPQUE
The stored procedure was stopped. SQL CALL requests for the stored procedure are queued until the stored procedure is started or the installation-defined timeout interval expires.

The -START PROCEDURE command can be used to place the stored procedure in STARTED status.

STOPREJ
The stored procedure was placed in this status by the STOP PROCEDURE ACTION(REJECT) command. SQL CALL requests for the stored procedure are rejected.

The -START PROCEDURE command can be used to place the stored procedure in STARTED status.

active The number of DB2 agents that are currently running this stored procedure.

qued The number of DB2 agents that are currently waiting for this stored procedure to be scheduled.

maxq The maximum number of DB2 agents that have waited for this stored procedure to be scheduled since DB2 was started.

timeout The number of times an SQL CALL statement timed out while waiting for a stored procedure to be scheduled since DB2 was started.

An SQL CALL statement can time out for several reasons:

- The procedure can be stopped by the STOP PROCEDURE ACTION(QUEUE) command, causing requests for the procedure to be queued until the installation-defined time interval expires.
- There might not be enough MVS TCBs in the DB2 stored procedures address space to satisfy the volume of SQL CALL statements.
- The MVS dispatching priority of the DB2 stored procedures address space might be too low to process the stored procedure requests in a timely manner.

fail The Number of times this procedure has abended since it was last started

wlm_env
The MVS Workload Manager environment name for this stored procedure.

System action: The DISPLAY PROCEDURE command processing continues.

DSNX943I PROCEDURES *begin* **-** *end* **STOP (***act***)**

Explanation: This message indicates a STOP PROCEDURE command was issued with an invalid procedure name, causing a range of procedure names to be stopped. This message follows the DSNX940I, DSNX941I, and any DSNX942I messages.

begin The first procedure name in the range of procedure names that are currently stopped

end The last procedure name in the range of procedure names that are currently stopped

act The action provided on the STOP PROCEDURE command

System action: The DISPLAY PROCEDURE command processing continues.

DSNX944I *csect-name* **THE STORED PROCEDURE FUNCTION IS NOT AVAILABLE**

Explanation: This message is returned by the START, STOP, and DISPLAY PROCEDURE command when the stored procedure facility is not available. It can also be produced when an MVS START command is used to start the stored procedures address space.

The following conditions cause the stored procedure facility to be unavailable:

- The DB2 DSNZPARM module does not include a value for the stored procedure JCL procedure name. This prevents DB2 from starting the stored procedure function.
- DB2 was unable to start the stored procedures address space. This can be caused by a JCL error in the stored procedures JCL procedure.
- DB2 is running on MVS/XA, which does not support the stored procedure function.

System action: If a START, STOP, or DISPLAY PROCEDURE command is in progress, the command terminates.

If an MVS START command is used to start the stored procedures address space, the stored procedures address space terminates.

DSNX945I DISPLAY PROCEDURE REPORT COMPLETE

Explanation: The output from the DISPLAY PROCEDURE command is complete. This message follows the DSNX940I, DSNX941I, and any DSNX942I messages.

System action: The DISPLAY PROCEDURE command processing continues.

DSNX946I *csect-name* **START PROCEDURE SUCCESSFUL FOR** *token*

Explanation: The START PROCEDURE command completed successfully.

token The procedure name supplied on the START PROCEDURE command.

System action: The START PROCEDURE COMMAND processing continues.

DSNX947I *csect-name* **STOP PROCEDURE SUCCESSFUL FOR** *token*

Explanation: The STOP PROCEDURE command completed successfully.

token The procedure name supplied on the STOP PROCEDURE command

System action: The STOP PROCEDURE command processing continues.

DSNX950I *csect-name* **PROCEDURE HAS NOT**
BEEN ACCESSED OR IS NOT
DEFINED

Explanation: The routine name supplied on the
DISPLAY PROCEDURE command either does not exist
or has not yet been accessed since DB2 started. The
specific procedure name might not be present in the
SYSIBM.SYSROUTINES catalog table.

System action: Processing continues for any additional procedure names supplied on the command.

DSNX951I *csect-name* **A STORED PROCEDURE CANNOT ISSUE START/STOP PROCEDURE COMMANDS**

Explanation: A stored procedure attempted to issue the START or STOP PROCEDURE command. This is not allowed because the START and STOP PROCEDURE commands might cause the stored procedures address space tasks to terminate.

System action: The START or STOP PROCEDURE command is rejected.

DSNX952I *csect-name* **THE DB2-Established STOREDPROCEDURE ADDRESS SPACE IS ALREADY STARTED**

Explanation: DB2 supports only one DB2-established stored procedure address space.

An MVS address space attempted to connect to DB2 as a DB2-established stored procedure address space, but another DB2-established stored procedure address space is already connected to DB2.

DB2 supports multiple WLM-established stored procedure address spaces.

System action: The attempt to connect to DB2 is rejected.

DSNX953E *csect-name* **STORED PROCEDURE JOBNAME OR STARTED TASK NAME DOES NOT MATCH ZPARM**

Explanation: An MVS address space attempted to identify itself to DB2 as the DB2-established stored procedure address space,but the JOBNAME or started task name of the address space did not match the stored procedure's MVS procedure name recorded in the DB2 DSNZPARM module.

System action: The attempt to connect to DB2 is rejected.

System programmer response: If you want to start the DB2-established stored procedure address space fromthe MVS console, you must use an MVS procedure name that matches the value in the DB2 DSNZPARM module.

DSNX954I *csect-name* **ATTEMPT TO START DB2-Established STORED PROCEDURE ADDRESS SPACE PROC=** *jclproc* **FAILED**

Explanation: DB2 attempted to start the DB2-established stored procedure address space using the JCL proc name specified in the DB2 DSNZPARM module. MVS failed the request.

Possible causes for the start failing include:

- JCL proc not found
- JCL error in the proc

jclproc The name of the JCL PROC used to start the stored procedures address space.

System action: The DB2-established stored procedures address space will not be available.

System programmer response: Correct the error and use the -START PROCEDURE(*) command to start the DB2-established stored procedure address space.

DSNX955I

Explanation: A blank line that appears in the DISPLAY PROCEDURE or DISPLAY FUNCTION command.

System action: The DISPLAY command processing continues.

DSNX956I **------ SCHEMA = schema**

Explanation: This message reports the schema name requested on the DISPLAY PROCEDURE or DISPLAY FUNCTION command.

schema The schema name for the stored procedure or function.

System action: The DISPLAY command processing continues.

DSNX957I *csect-name* **INSUFFICIENT AUTHORITY FOR OBJECT** *object*

Explanation: This message is issued when a START, STOP, or DISPLAY command is issued for a stored procedure or user-defined function that the user is not authorized to START, STOP, or DISPLAY.

object The stored procedure or user-defined function that failed the authorization test.

System action: The START, STOP, or DISPLAY command continues.

DSNX958I **DISPLAY TERMINATED WITH MAX LINES**

Explanation: This message indicates that the -DISPLAY PROCEDURE or -DISPLAY FUNCTION SPECIFIC report was truncated. The report was generated in response to a command from a MVS console, and more than 254 response messages wree generated. Only 254 response messages are returned.

System action: Command processing completes. The truncated report is written to the MVS console.

System programmer response: Reissue the -DISPLAY PROCEDURE or -DISPLAY FUNCTION SPECIFIC command, specifying a smaller number of routines. This can be done by specifying a list of routines, a schema qualifier, or a partial routine name in the

DISPLAY command rather than the default (*.*). Alternatively, the command can be reissued from the DB2I Commands panel or from a batch job rather than from the MVS console.

DSNX960E *csect-name* **SOME SQL CALL STATEMENTS CANNOT BE PROCESSED, BECAUSE THE Language Environment ROUTINE CEEPIPI CANNOT BE LOADED SSN=** *ssn* **PROC=** *jclproc* **ASID=** *asid* **WLM_ENV=** *wlmenv*

Explanation: To execute an SQL CALL statment, DB2 must load the Language Environment interface module (CEEPIPI). If the CEEPIPI module is not included in an accessible data set such as STEPLIB data sets, JOBLIB data sets, or the LPA library, DB2 is unable to call Language Environment to process stored procedure or function requests.

ssn The name of the DB2 subsystem to which the stored procedures address space was connected.

ssn The name of the JCL PROC used to start this stored procedures address space.

asid The MVS ASID of the address space where the load was issued.

wlmenv The name of the WLM application environment where the load was issued. If it executed in the DB2-established stored procedures address space, this field will contain blanks.

System action: The address space attempting to process stored procedures or functions terminates.

System programmer response: Change the STEPLIB DD statement in the JCL used to start the stored procedure address space, or install the CEEPIPI module in one of the libraries already accessible to the address space.

After the CEEPIPI module is accessible, use one of the following techniques to restart the failing stored procedure address space:

- If the error was in a WLM-established stored procedures address space, use the MVS VARY WLM,APPLENV=name,RESUME command.
- If the error was in the DB2-established stored procedures address space, use the -START PROCEDURE(*) command.

Problem determination: Collect the following diagnostic items listed in Appendix A, “Problem determination,” on page 659: 1.

DSNX961I *csect-name* **ATTEMPT TO PERFORM JNI FUNCTION** *function* **FAILED FOR STORED PROCEDURE** *procedure-name* **SSN=***ssn* **PROC=***jclproc* **ASID=***asid* **CLASS=***class-name* **METHOD=***method-name* **ERROR INFO=** *error-information*

Explanation: A WLM-established JAVA stored procedures address space could not perform a necessary function. The following list contains the possible values for each variable in the message:

function
The function that failed. The following list contains the functions that could have failed:

- CallStaticMethod
- CreateJavaVM
- DestroyJavaVM
- FindClass
- INTERNAL_ERROR
- JDBC Reset
- QueryJavaVM

procedure-name
The name of the stored procedure that was invoked.

ssn The name of the DB2 subsystem to which the stored procedures address space is connected.

jclproc
The name of the JCL procedure that was used to start the stored procedures address space.

asid
The ASID of the address space in which *function* was performed.

class-name
The name of the class that contains the Java method to be invoked.

method-name
The name of the Java method to be invoked.

error-information
The error information provided for this error.

System action: The CALL statement fails.

System programmer response: Use the value of *function* and this message’s *error-information* to correct the problem. If the WLM-managed stored procedure address space is running with a //JSPDEBUG DD card, then, for further information about this error, check its output (typically, the WLM-managed stored procedure address space output) for information.

Also, check the HFS directory identified as the home directory for the ID associated with the WLM-managed stored procedure address space for any dumps that the JVM could have taken. Check either the `/tmp/java` directory, or if WORK_DIR is specified then its

directory, for information that could have been written to file `server_stderr.txt`.

CallStaticMethod
This *function* indicates that an exception has taken place that was not a Java SQLException.

CreateJavaVM
This *function* indicates that the JVM could not be created. You must have the IBM Developer Kit for OS/390, Java 2 Technology Edition, and the JAVA_HOME environmental variable in your JAVAENV dataset must reference its home directory.

DestroyJavaVM
This *function* indicates that the JVM could not be destroyed without error.

FindClass
This *function* indicates that either a required system-provided class could not be found or an error occurred in accessing a system- or user-provided class. You need to take a specific action based on what type of class could not be found or accessed:

- For a system-provided class, check the settings for JAVA_HOME, and DB2_HOME or JCC_HOME in the JAVAENV DD statement in the stored procedures address space start-up procedure, and ensure that they are set correctly.
- For a user-provided class, check the settings for TMSUFFIX and CLASSPATH in the JAVAENV DD statement in the stored procedures address space start-up procedure, and check the routine's use of an installed JAR. The JVM runtime loads all classes that are referenced in the identified class. Ensure that all referenced, user-provided, classes are in the CLASSPATH or TMSUFFIX, or in a referenced installed JAR.

INTERNAL_ERROR
This *function* indicates that an internal error has occurred. Record the following information and contact your IBM Support Center for assistance:

- Any information reported in this message's *error-information*
- The results of the recommended checks of the WLM-managed stored procedure address space
- Any dumps that the JVM could have taken
- Any information written to file `server_stderr.txt`

JDBC Reset
This *function* indicates that a specific internal error has occurred. Record the following information and contact your IBM Support Center for assistance:

- Any information reported in this message's *error-information*
- The results of the recommended checks of the WLM-managed stored procedure address space
- Any dumps that the JVM could have taken
- Any information written to file `server_stderr.txt`

QueryJavaVM
This *function* indicates that the JVM is being destroyed and recreated. The value given for *error-information* will indicate that either a status query of the JVM failed, or, the JVM is unresettable.

DSNX962I *csect-name* **THE Language Environment CEEPIPI SERVICE RETURNED AN UNEXPECTED RETURN CODE** *rc* **FOR FUNCTION** *func* **SSN=** *ssn* **PROC=** *jclproc* **ASID=** *asid* **WLM_ENV=** *wlmenv*

Explanation: DB2 uses the Language Environment CEEPIPI service to load and execute user-written stored procedures. The Language Environment return code *rc* was not expected for the indicated Language Environment function call.

rc The return code received from CEEPIPI.

func The function requested by DB2. Possible values are:

add_entry
Add an entry to the Language Environment PIPI table dynamically.

call_main
Run a main program stored procedure using an existing Language Environment run time environment.

call_sub
Run a sub program stored procedure using an existing Language Environment run time environment.

delete_entry
Delete an entry from the Language Environment PIPI table dynamically.

end_seq
End a sequence of calls using CEEPIPI.

identify
Identify the language the module was written in.

init_main
Create Language Environment run time environment.

init_sub_dp
Create Language Environment sub run time environment.

term Terminate the Language Environment run time environment.

start_seq
Start a sequence of calls using CEEPIPI.

ssn The name of the DB2 subsystem to which the stored procedures address space was connected.

jclproc The name of the JCL PROC used to start this stored procedures address space.

asid The MVS ASID of the address space where the CEEPIPI error was encountered.

wlmenv The name of the WLM application environment where the CEEPIPI error was encountered. If it executed in the DB2-established stored procedures address space, this field will contain blanks.

System action: The attempt to run a stored procedure fails. The DB2 stored procedure address space attempts any new stored procedure requests it receives.

System programmer response: The Language Environment return codes from CEEPIPI are documented in *Language Environment for OS/390 & VM Programming Guide*.

If the values of *func* and *rc* are 'ADD_ENTRY' and X'0000000C', the error might be caused by using a compiler that is not Language Environment enabled to create the stored procedure. DB2 stored procedures must be compiled with a compiler that supports Language Environment.

Problem determination: Collect the following diagnostic item listed in Appendix A, "Problem determination," on page 659: 1.

DSNX964I *csect-name* **THE DB2-EstablishedSTORED PROCEDURES ADDRESS SPACE FOR SUBSYSTEM** *subs* **IS STARTED**

Explanation: The DB2-established stored procedures address space is now available to process SQL CALL statements.

subs The name of the DB2 subsystem to which the DB2-established stored procedures address space is connected.

System action: Processing continues.

DSNX965I *csect-name* **THE DB2-Established STORED PROCEDURES ADDRESS SPACE FOR SUBSYSTEM** *subs* **IS STOPPING**

Explanation: The DB2-established stored procedures address space is terminating. Termination occurs when all active stored procedures in the DB2-established stored procedures address space return control to DB2.

subs The name of the DB2 subsystem to which the DB2-established stored procedures address space was connected.

System action: DB2 continues to process any stored procedures that are currently running. If a new SQL CALL statement is received that is to be executed in the DB2-established stored procedures address space the request is queued.

DSNX966I *csect-name* **STORED PROCEDURE OR FUNCTION** *sp* **FAILED. Language Environment ENCLAVE RC =** *erc*, **FEEDBACK INFORMATION =** *fi* **SSN=** *ssn* **PROC=** *jclproc* **ASID=** *asid* **WLM_ENV=** *wlmenv*

Explanation: DB2 uses the Language Environment CEEPIPI call_main or call_sub service to execute user-written stored procedures or user-defined function. The CEEPIPI interface returned information indicating a failure while executing the stored procedure.

sp The stored procedure or function name that was called using CEEPIPI.

erc The Language Environment enclave return code.

fi Contains one or two message identifiers. The message identifiers are constructed from the Language Environment condition token.

The message identifiers are constructed as follows:
- Single message: Facility_Id combined with Msg_No
- Two messages: Facility_Id combined with Class_Code and Facility_Id combined with Cause_Code

ssn The name of the DB2 subsystem that called the stored procedure or function.

jclproc The name of the JCL PROC used to start this stored procedures address space.

asid The MVS ASID of the address space where the stored procedure executed.

wlmenv The name of the WLM application environment where the stored procedure executed. If it executed in the DB2-established stored procedures address space, this field will contain blanks.

System action: The attempt to run this stored procedure failed. The DB2 stored procedure address space attempts any new stored procedure requests it receives.

System programmer response: A description of the Language Environment enclave return code and the condition token are documented in *Language Environment for OS/390 & VM Programming Guide*. In many cases the feedback information contains a

Language Environment message number. These are documented in z/OS Language Environment.

Problem determination: Collect the following diagnostic item listed in Appendix A, "Problem determination," on page 659: 1. 4.

DSNX967I *csect-name* **ATTEMPT TO PERFORM WLM FUNCTION** *func* **FAILED WITH WLM RC=** *rc* **RSN=** *rsn* **SSN=** *ssn* **PROC=** *jclproc* **ASID=** *asid* **WLM_ENV=** *wlmenv*

Explanation: A WLM-established stored procedures address space was unable to perform a WLM operation required to communicate with WLM.

func The WLM function that failed. Possible values
are:
• IWMCONN
• IWMSSEL
• IWMSTBGN
• IWMSTEND

If the value for *func* is IWMSTEND with WLM
RC=00000008 RSN=0033E0859, then the error
is caused by extra JVM threads that are left
behind after a Java routine executed. You can
take one of the following actions:

• Ignore this error message, which means that
the JVM will be destroyed and re-initialized
each time the routine runs.

• Add a NOWLMENC DD statement to the
JCL for the WLM-SPAS where this routine
executes. This will cause the routine to be
executed outside the WLM enclave.

ssn The name of the DB2 subsystem to which this stored procedure address space was connected.

jclproc The name of the JCL PROC used to start this stored procedures address space.

asid The MVS ASID of the address space where this function was invoked.

wlmenv The name of the WLM application environment where this function was invoked.

System action: The stored procedures service task that encountered the error terminates. If this is the last service task in that address space, then that address space terminates.

The WLM return codes and reason codes are documented in *z/OS MVS Programming: Workload Management Services*. In some cases the failures are caused by the WLM application environment being incorrectly defined or in a quiesced state. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 4, 5.

DSNX968I *csect-name* **STORED PROCEDURE ADDRESS SPACE IS UNABLE TO CONNECT TO WLM BECAUSE WLM_ENV =** *wlmenv* **IS STOPPED OR QUIESCED**

Explanation: A DB2 stored procedures address space could not connect to the Work Load Manager services of MVS because the WLM application environment was in a stopped or quiesced state.

wlmenv The name of the WLM application environment for which the stored procedure address space was being started.

System action: The stored procedures address space terminates.

System programmer response: Correct the WLM application environment name that is being used by stored procedures address space JCL proc.

To change the status of the application environment use the MVS VARY WLM,APPLENV=wlmenv,RESUME command.

If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 4, 5.

DSNX969I *function status active qued maxq timeout fail*
wlm_env

Explanation: This message lists the DISPLAY SPECIFIC FUNCTION information for a given external function. The definitions of the columns are:

function The name of the external function.

status The status of the function. The status can be any of the following values:

STARTED
The function is able to accept SQL statement requests.

STOPABN
The function was stopped because of the previous abnormal termination of the function application. SQL statment requests for the function are rejected. The -START SPECIFIC FUNCTION command can be used to place the function in STARTED status.

STOPQUE
The function was stopped. SQL statement requests for the function are queued until the function is started or the installation-defined

timeout interval expires. The -START SPECIFIC FUNCTION command can be used to place the function in STARTED status.

STOPREJ
The function was placed in this status by the STOP SPECIFIC FUNCTION ACTION(REJECT) command. SQL statement requests for the function are rejected. The -START SPECIFIC FUNCTION command can be used to place the function in STARTED status.

active The number of DB2 agents that are currently running this function.

qued The number of DB2 agents that are currently
waiting for this function to be scheduled.

maxq The maximum number of DB2 agents that
have waited for this function to be scheduled,
since the last START SPECIFIC FUNCTION
command was issued for this function.

timeout
The number of times an SQL statement timed out while waiting for a function to be scheduled since the last START SPECIFIC FUNCTION command was issued for this function. An SQL statement can time out because the function can be stopped by the STOP SPECIFIC FUNCTION ACTION(QUEUE) command, causing requests for the function to be queued until the installation-defined time interval expires.

fail The Number of times this function has
abended since it was last started.

wlm_env
The MVS Workload Manager environment name for this function.

System action: The DISPLAY SPECIFIC FUNCTION command processing continues.

DSNX970I FUNCTION STATUS ACTIVE QUED
MAXQ TIMEOUT FAIL WLM_ENV

Explanation: This message displays the column headings for the list of external functions generated by the DISPLAY SPECIFIC FUNCTION command. The definitions are:

FUNCTION
The name of the external function.

STATUS
The status of the function.

ACTIVE
The number of DB2 agents that are currently running this function.

QUED The number of DB2 agents that are currently
waiting for this function to be scheduled.

MAXQ The maximum number of DB2 agents that
have waited for this function to be scheduled,
since the last START SPECIFIC FUNCTION
command was issued for this function.

TIMEOUT
The number of times an SQL statement timed out while waiting for a function to be scheduled since the last START SPECIFIC FUNCTION command was issued for this function.

FAIL The Number of times this function has
abended since it was last started.

WLM_ENV
The MVS Workload Manager environment name for this function.

System action: The DISPLAY SPECIFIC FUNCTION command processing continues.

DSNX971I FUNCTIONS *begin* **-** *end* **STOP (***act***)**

Explanation: This message indicates a STOP SPECIFIC FUNCTION command was issued with an asterisk ('*') in the function name, causing a range of function names to be stopped. This message follows the DSNX975I, DSNX969I, and any DSNX970I messages.

begin The first function name in the range of function names that are currently stopped

end The last function name in the range of function names that are currently stopped

act The action provided on the STOP SPECIFIC FUNCTION command

System action: The DISPLAY SPECIFIC FUNCTION command processing continues.

DSNX972I DISPLAY FUNCTION SPECIFIC REPORT COMPLETE

Explanation: The output from the DISPLAY FUNCTION SPECIFIC command is complete. This message follows DSNX975I, DSN969I, and any DSNX970I messages.

System action: The DISPLAY FUNCTION SPECIFIC command processing continues.

DSNX974I *csect-name* **STOP FUNCTION SPECIFIC SUCESSFUL FOR** *token*

Explanation: The function name supplied on the STOP FUNCTION SPECIFIC command completed sucessfully.

token The function name supplied on the STOP FUNCTION SPECIFIC command.

System action: The STOP FUNCTION SPECIFIC

command processing continues.

DSNX975I *csect-name* **- DISPLAY FUNCTION SPECIFIC REPORT FOLLOWS-**

Explanation: This message is the title for the report output from the DISPLAY FUNCTION SPECIFIC command. It precedes messages DSNX969I and DSNX970I. Message DSNX971I might also be displayed. The final message for the report is DSNX972I.

System action: The DISPLAY FUNCTION SPECIFIC command processing continues.

DSNX977I *csect-name* **FUNCTION HAS NOT BEEN**
ACCESSED OR IS NOT DEFINED

Explanation: The routine name supplied on the
DISPLAY PROCEDURE command either does not exist
or has not yet been accessed since DB2 started. The
specific function name might not be present in the
SYSIBM.SYSROUTINES catalog table.

System action: Processing continues for any additional function names supplied on the command.

DSNX978I **DISPLAY TERMINATED DUE TO INSUFFICIENT MESSAGE SPACE**

Explanation: The DISPLAY PROCEDURE command could not obtain storage for additional messages. This should only happen for a particularly long display request, such as one requesting all stored procedures that have been accessed by DB2 applications since DB2 was started.

System action: The display is truncated.

System programmer response: Reissue the -DISPLAY PROCEDURE command, and specify a smaller number of procedures. This can be done by specifying a partial procedure name in the DISPLAY PROCEDURE command rather than the default (*).

DSNX980E *csect-name* **ATTEMPT TO PERFORM CALL ATTACH FUNCTION** *func* **FAILED WITH DB2 SUBSYSTEM** *subsys***. RETURN CODE=***rc***, REASON CODE=***reason*

Explanation: The DB2 stored procedures address space was unable to perform a DB2 call attach operation required to communicate with DB2.

func The DB2 call attach function that failed. Possible values are:
CONNECT
OPEN
CLOSE
DISCONNECT

subsys The DB2 subsystem name involved in the call attach function

rc The DB2 call attach return code

reason The DB2 reason code

System action: The stored procedures service task that encountered the error terminates. If this is the last service task in the address space, the the DB2 stored procedures address space terminates.

System programmer response: Consult the DB2 *DB2 Application Programming and SQL Guide* to determine the cause of the DB2 call attach return code and reason code.

DSNX981E *csect-name* **THE PARAMETER** *parm* **CONTAINS AN INVALID VALUE** *"string"* **PROC=** *jclproc*

Explanation: The DB2 stored procedures address space received an invalid value in the JCL parameter used to start the address space.

parm The JCL parameter is specified incorrectly. Possible values are:

SUBSYS
This parameter can be invalid for the following reasons:
- The SUBSYS parameter is null.
- The SUBSYS parameter is too long.
- The SUBSYS parameter contained an invalid character.

NUMTCB
This parameter can be invalid for the following reasons:
- A comma was not present in the parameter list to separate the NUMTCB parameter from the previous parameter.
- The NUMTCB parameter is null.
- The NUMTCB parameter contained nonnumeric characters.
- The value of the NUMTCB parameter is zero.
- The value of the NUMTCB parameter is negative.
- The value of the NUMTCB parameter is greater than the maximum number of TCBs allowed in the DB2 stored procedures address space.

APPLENV
This parameter can be invalid for the following reasons:
- A comma was not present in the parameter list to separate the

APPLENV parameter from the previous parameter.
- The value of the APPLENV parameter is longer than eighteen characters.

string The value provided for the parameter in error.

jclproc The name of the JCL PROC used to start this stored procedures address space.

System action: The DB2 stored procedures address space terminates.

If MVS WLM is used to establish the stored procedure address space, WLM will retry the start of the address space two more times. If all three attempts fail, WLM will quiesce the WLM environment associated with the stored procedure address space.

System programmer response: Correct the JCL parameter used to invoke the DB2 stored procedure address space.

If MVS WLM is used to establish the stored procedure address space, the MVS VARY WLM,APPLENV=*wlmenv*,RESUME command might be required to resume the WLM application environment (*wlmenv*) associated with the stored procedures address space.

DSNX982I *csect-neme* **ATTEMPT TO PERFORM RRS ATTACH FUNCTION** *func* **FAILED WITH RRS ATTACH RC=** *rc* **RSN=** *rsn* **SSN=** *ssn* **PROC =** *jclproc* **ASID=** *asid* **WLM_ENV=** *wlmenv*

Explanation: A WLM-established stored procedures address space was unable to perform an RRS attach operation required to communicate with DB2.

func The RRS attach function that failed. Possible values are:
- SPAS_ID
- SPAS_INIT_SP
- SPAS_TERM_SP

ssn The name of the DB2 subsystem involved with the RRS attach function.

jclproc The name of the JCL PROC used to start the stored procedures address space.

asid The MVS ASID of the address space where the function was attempted.

wlmenv The name of the WLM application environment where the function was attempted.

System action: The stored procedures service task that encountered the error terminates. If this is the last service task in that address space, then that address space terminates.

System programmer response: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination:

Collect the following diagnostic items listed in Appendix A, “Problem determination,” on page 659: 1, 4, 5.

DSNX983I *csect-neme* **DB2 STORED PROCEDURE SUPPORT CODE AND SUBSYSTEM CODE ARE NOT AT THE SAME LEVEL SSN=** *ssn* **PROC =** *jclproc*

Explanation: A DB2 stored procedures address space could not connect to the DB2 subsystem. This occurred because the DB2 code in the stored procedures address space was not at the same release level as the code in the DB2 subsystem.

ssn The name of the DB2 subsystem that the stored procedure address space was attempting to connect with.

jclproc The name of the JCL PROC used to start the stored procedures address space.

System action: The stored procedures address space terminates.

System programmer response: Correct the DB2 library name in the STEPLIB for the JCL proc for the stored procedures address space. Make the library name the same as the library name used for the DB2 subsystem address spaces. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: Collect the following diagnostic items listed in Appendix A, “Problem determination,” on page 659: 1, 4, 5.

DSNX984I *csect-name* **ALL FUNCTIONS STARTED FOR THIS SCHEMA.**

Explanation: All functions are started for the schema listed in the preceeding DSNX956I message. This message is issued only when STOP FUNCTION SPECIFIC(*.*) was in effect.

System action: The DISPLAY SPECIFIC FUNCTION command processing continues.

DSNX985I *csect-name* **ALL PROCEDURES STARTED FOR THIS SCHEMA.**

Explanation: All procedures are started for the schema listed in the preceeding DSNX956I message. This message is issued only when a STOP PROCEDURE(*.*) was in effect.

System action: The DISPLAY PROCEDURE command processing continues.

DSNX986I **ATTEMPT TO** *func* **FAILED WITH RV =** *rv* **RC =** *rc* **RSN =** *rsn* **SSN =** *ssn* **PROC =** *jclproc*

Explanation: A WLM-established stored procedures address space was unable to register itself to z/OS UNIX System Services. z/OS UNIX System Services are not available to stored procedures running in the address space.

func The dub setting used by z/OS UNIX System Services BPX1SDD callable service. This can be DUBPROCESS or DUBTHREAD.

rv The return value given by z/OS UNIX System Services BPX1SDD callable service.

rc The return code given by z/OS UNIX System Services BPX1SDD callable service.

rsn The reason code given by z/OS UNIX System Services BPX1SDD callable service.

ssn The name of the DB2 subsystem to which this stored procedure address space was connected.

proc The name of the JCL PROC used to start this stored procedures address space.

System action: The stored procedure address space initialization continues. z/OS UNIX System Services callable services' return values, return codes, and reason codes are documented in *z/OS UNIX System Services Messages and Codes*.

Problem determination: Most failures are due to the user ID associated with the WLM-established stored procedures address space not being RACF defined. If you suspect an error in DB2, refer to Part 3 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

It is also possible that the number of processes was exceeded for the user ID. If the NUMBER OF TCBS is set too high during install, NUMTCB may exceed the value of the UNIX System Services MACPROCUSER setting. Refer to *z/OS UNIX System Services Messages and Codes* for more information.

DSNX987I *csect-name* **NAME** *schema-name* **INVALID, COMMAND IGNORED.**

Explanation: The function name or procedure name supplied on the DISPLAY, START, or STOP, FUNCTION SPECIFIC or PROCEDURE command uses a invalid wildcard schema name. The qualified routine name must specify a full schema name, "*", "*.*", or blanks.

schema-name
The schema name containing invalid characters.

System action: The START, STOP, or DISPLAY command is rejected.

DSNX988I *csect-name* **ALL OTHER FUNCTIONS IN OTHER SCHEMAS STOPPED** *act*

Explanation: A STOP FUNCTION SPECIFIC(*.*) is currently in effect. Any functions not listed in the preceeding DSNX969, DSNX971, and DSNX976 messages are stopped *act*.

act The action provided on the STOP FUNCTION command

System action: The DISPLAY SPECIFIC FUNCTION command processing continues.

DSNX989I *csect-name* **ALL OTHER PROCEDURES IN OTHER SCHEMAS STOPPED** *act*

Explanation: A STOP PROCEDURE(*.*) is currently in effect. Any functions not listed in the preceeding DSNX942, DSNX943, and DSNX950 messages are stopped *act*.

act The action provided on the STOP PROCEDURE command

System action: The DISPLAY PROCEDURE command processing continues.

DSNX990E *csect-name* **IMS ODBA INITIALIZATION FAILED BECAUSE THE CALLABLE INTERFACE CANNOT BE LOADED**

Explanation: The IMS ODBA callable interface module cannot be loaded. No IMS database DL/I calls will be supported from stored procedures. Ensure the load module named AERTDLI is located within the partitioned dataset specified in the stored procedure address space startup procedure's DFSRESLB dd card statement.

System action: The stored procedure address space initialization continues.

DSNX991I *csect-name* **IMS ODBA INITIALIZATION COMPLETED**

Explanation: The initialization of the IMS ODBA callable interface has completed. Stored procedure applications may use DL/I calls to access IMS databases.

System action: The stored procedure address space initialization continues.

DSNX992E *csect-name* **IMS ODBA INITIALIZATION FAILURE, AIB RC =** *rc*

Explanation: The initialization of the IMS ODBA callable interface has failed. Stored procedure applications cannot use DL/I calls to access IMS databases.

rc The AIB return code provided by IMS for the CIMS INIT call. Refer to the section on DL/I Return and Reason Codes in *IMS Application Programming: Database Manager* for more information regarding the failure.

System action: The stored procedure address space initialization continues.

DSNX993I *csest-name* **CALL TO REXX PROCEDURE** *execname* **FAILED, FUNCTION =** *function* **RC =** *rc* **RSN =** *rsn*

Explanation: An attempt to establish an environment to run a REXX stored procedure has failed. Use the following pieces of information to identify the error:

execname
The name of the REXX exec.

function The TSO or REXX function call that failed. It can be one of three values: IKJTSOEV, IRXINIT, or IRXEXEC. For more information regarding the IKJTSOEV function, refer to *z/OS TSO/E Programming Services*. For information regarding IRXINIT and IRXEXEC, refer to *z/OS TSO/E REXX Reference*

rc The return code for the specific function, provided by TSO or REXX. For return code explanations, refer to *z/OS TSO/E Programming Services* and *z/OS TSO/E REXX Reference*.

reason The reason code for the specified function, provided by TSO or REXX. For reason code explanations, refer to *z/OS TSO/E Programming Services* and *z/OS TSO/E REXX Reference*.

System action: The stored procedure address space initialization continues.

Problem determination: View the file associated with the SYSTSPRT DD card in the Stored Procedure Address Space start-up procedure. This file contains diagnostic messages issued by TSO regarding the failing function. Refer to *z/OS TSO/E Messages* for any messages found in the SYSTSPRT.

DSNX994I SCOPE(GROUP) COMMAND CANNOT BE PROCESSED DUE TO COMMAND ALREADY IN PROGRESS

Explanation: A -DISPLAY PROCEDURE, -START PROCEDURE, -STOP PROCEDURE, -DISPLAY FUNCTION, -START FUNCTION, or -STOP FUNCTION command with SCOPE(GROUP) was issued at the same time that another data sharing member was processing a SCOPE(GROUP) command. DB2 could not obtain the lock necessary to serialize these commands.

Severity: 8 (error)

System action: The command is not processed on any members of the data sharing group.

User response: Enter the command again.

Chapter 22. DSNY... messages

DSNY001I SUBSYSTEM STARTING

Explanation: The -START DB2 command is accepted. This message is issued to the MVS console that issued the -START command. The DSN9022I message is sent when the DB2 subsystem startup process has completed. Refer to the DSN9022I message for additional information.

System action: DB2 startup processing begins.

DSNY002I SUBSYSTEM STOPPING

Explanation: The -STOP DB2 command is accepted. The DSN9022I message is issued when the DB2 subsystem shutdown process has completed. The message is issued either to the origin of the -STOP DB2 command (for MVS MCS console) or to the MVS console from which the -START DB2 command was received. Refer to the DSN9022I message for additional information.

System action: DB2 subsystem shutdown is initiated.

DSNY003I SUBSYSTEM IS ALREADY ACTIVE

Explanation: The -START DB2 command has not been accepted, because the DB2 subsystem is active. The DSN9023I message is issued after this message. Refer to the DSN9023I message for additional information.

DSNY004I SUBSYSTEM IS ALREADY STOPPING

Explanation: The -STOP DB2 command has not been accepted either because the DB2 subsystem shutdown is in progress for the specified option (QUIESCE or FORCE), or because the QUIESCE option was specified after a FORCE option had been previously accepted. The DSN9023I message is issued after this message. Refer to the DSN9023I message for additional information.

System action: DB2 subsystem shutdown is continued.

DSNY005I SUBSYSTEM STARTUP TERMINATED, INVALID START COMMAND

Explanation: The DB2 subsystem can be started only by a -START DB2 command.

System action: DB2 subsystem startup is terminated.

Operator response: Start the DB2 subsystem via the -START DB2 command, and reenter the rejected command.

DSNY006I *csect-name* **INVALID AMODE OR RMODE ATTRIBUTE FOUND FOR LOAD MODULE** *module name*

Explanation: The Initialization Procedures subcomponent detected that a module had an invalid AMODE or RMODE attribute when it was loaded. In the error message, 'module name' is the name of the load module with an invalid addressing or residency mode.

System action: DB2 subsystem startup is terminated.

Operator response: Notify the system programmer of the failure.

System programmer response: Verify that all installation and maintenance activities against DB2 were executed using SMP, that the JCLIN for SMP includes the correct AMODE and RMODE control statements, and that the Linkage Editor provided with the Data Facility Product (DFP) was invoked by SMP for the link edits. Correct the procedure or JCL that caused the error and rerun the necessary SMP jobs.

Problem determination: A DB2 dump was requested to an MVS SYS1.DUMP data set. The MVS service aid AMBLIST provides Linkage Editor module attributes in the summary section of the LISTLOAD function output.

DSNY007I LOAD MODULE DSNHDECP DOES NOT HAVE RMODE(24) ATTRIBUTE

Explanation: The named load module was not link-edited with the required RMODE(24) attribute.

System action: DB2 startup is abended with reason code 00E80160.

Operator response: Notify the system programmer.

System programmer response: Verify that all installation and maintenance activities against DSNHDECP were executed using SMP. JCLIN for SMP must have included the correct AMODE and RMODE control statements. The Linkage Editor, provided with the Data Facility Product (DFP), was invoked by SMP for the link-edits. Correct the procedure or JCL that caused the error and rerun the necessary SMP jobs.

Problem determination: The MVS service aid AMBLIST provides Linkage Editor module attributes in the summary section of the LISTLOAD function output.

DSNY008I SUBSYSTEM SHUTDOWN REQUEST NOT ACCEPTED

Explanation: The -STOP DB2 command has not been accepted because a subsystem startup has not

© Copyright IBM Corp. 1983, 2008

completed to the point where shutdown can occur. The DSN9023I message is issued after this message. Refer to the DSN9023I message for additional information.

System action: DB2 startup continues and the -STOP DB2 command is ignored.

User response: Reissue the -STOP DB2 command after subsystem startup has completed.

DSNY009I *csect-name* **SUBSYSTEM STARTING IN LIGHT MODE, NORMAL TERMINATION TO FOLLOW RELEASE OF RETAINED LOCKS IF NO INDOUBTS**

Explanation: DB2 issues this message during DB2 startup if it is restarted in light mode.

System action: DB2 startup continues normally. The DB2 member will automatically shut itself at the end of restart after the retained locks are removed if there are no indoubt units of recovery. If indoubt units of recovery exist, the DB2 member remains up and running in light mode so that the indoubts can be resolved.

DSNY010I **LOAD MODULE module name IS NOT AT THE CORRECT RELEASE/ VERSION LEVEL**

Explanation: The named load module is not at the correct release/version level for the version of DB2 that was being started.

System action: DB2 startup is abended with reason code 00E80161.

Operator response: Notify the system programmer.

System programmer response: Verify that the correct load libraries are being used. Also verify that the application of maintenance did not result in modules being link edited into the wrong load libraries.

| **DSNY011I** *csect-name* **Z/ARCHITECTURE WITH 64-BIT ADDRESSING IS REQUIRED FOR THIS RELEASE OF DB2**

| **Explanation:** An attempt was made to start DB2 on a processor that either does not support z/Architecture, or is not enabled for 64-bit addressing.

| **System action:** DB2 startup abends with reason code 00E8005A.

| **System programmer response:** Ensure that the processor on which DB2 is started supports z/Architecture and is enabled for 64-bit addressing.

| **DSNY012I** *csect-name***THIS RELEASE OF DB2 REQUIRES Z/OS V1R3 OR ABOVE**

| **Explanation:** An attempt was made to start DB2 on an operating system image that was not z/OS V1R3 or above.

| **System action:** DB2 startup abends with reason code 00E80058.

| **System programmer response:** Ensure that z/OS V1R3 or above is enabled before starting DB2.

| **DSNY014I** *n csect* **DB2 HAS BEEN STARTED WITH ACCESS(MAINT)**

| **Explanation:** This DB2 has been started with option ACCESS(MAINT).

| **Severity:** 0 (informational)

| **System action:** None.

| **Programmer response:** None.

DSNY015I **LIGHT(YES) ON START DB2 COMMAND WAS IGNORED, SYSTEM IS NOT ENABLED FOR DATA SHARING**

Explanation: DB2 issues this message during startup if DB2 is started with the LIGHT(YES) keyword and the system is not enabled for data sharing.

System action: DB2 startup continues normally, ignoring the LIGHT(YES) keyword.

DSNY020I *csect-name* **MEASURED USAGE DATA IS NOT BEING RECORDED FOR** *subsystem/address-space*

Explanation: For measured usage data to be recorded, SMF type 89 records must be collected. Two PTFs must be applied: one to the early code and one to the base. Until both are active, measured usage data cannot be recorded.

System action: Processing continues, but measured usage for either the DB2 subsystem or the specified DB2 address space is not recorded.

Programmer response: If measured usage data is needed, be sure that SMF type 89 records are collected and that both PTFs are applied to the running system. An IPL is required for the change to early code.

If the records are being recorded, the PTFs are applied, and an address-space is in the message, then the return code for the MVS service is located in ASCEMULC for that address space.

Chapter 23. DSNZ... messages

DSNZ002I **src DSNZINIT SUBSYSTEM ssnm SYSTEM PARAMETERS LOAD MODULE NAME IS dsnzparm-name**

Explanation: DB2 subsystem 'ssnm' is being started with the 'dsnzparm-name' DSNZPARM load module. This message is written to the MVS console which entered the -START DB2 command. The message will also appear in the JES SYSLOG and on the JES job log of *ssnm*MSTR. The message contains the name of the DSNZPARM load module entered by the operator or the default name (DSNZPARM) if no name was entered.

System action: DB2 startup processing continues.

DSNZ003I *csect-name* **DSNZINIT CSECT** *parm-csect* **NOT FOUND IN SYSTEM PARAMETERS.**

Explanation: The named *parm-csect* was not found in the system parameters (ZPARM) specified when starting DB2. This CSECT is required to start DB2. The name of the system parameters load module that is missing this CSECT is identified in previously issued message DSNZ002I.

This error can occur if the system parameters load module specified is for a different release of DB2.

System action: DB2 subsystem startup processing stops.

Operator response: Verify that the correct system parameters name was specified. If the correct name was specified, notify the system programmer.

System programmer response: Verify that:

- The system parameters load module is for the release of DB2 being started.
- The correct load libraries are being used to load the system parameters load module.

If these are correct, then reassemble and relink-edit your system parameters load module.

DSNZ004I *csect-name* **DSNZINIT INCORRECT LEVEL** *wrong-level* **FOUND IN** *parm-csect* **SYSTEM PARAMETERS. LEVEL** *correct-level* **IS REQUIRED.**

Explanation: The system parameters load module CSECT *parm-csect* contains an incorrect level identifier. The CSECT contains a level identifier of *wrong-level* and the DB2 that is starting requires a level identifier of *correct-level*.

This error can occur if the system parameters load module specified is for a different release of DB2. This error can also occur if the system parameters load module is missing some maintenance. In this case the *parm-csect* is down level and must be reassembled.

System action: DB2 subsystem startup processing stops.

Operator response: Verify that the correct system parameters name was specified. If the correct name was specified, then notify the system programmer.

System programmer response: Verify that:

- The system parameters load module is for the release of DB2 being started.
- The correct load libraries are being used to load the system parameters load module.

If these are correct, then reassemble and relink-edit your system parameters load module.

DSNZ006I *src* **SUBSYS** *ssnm* **SYSTEM PARAMETERS LOAD MODULE NAME** *module-name* **IS BEING LOADED**

Explanation: DB2 subsystem *ssnm* is loading parameters from *module-name* subsystem parameters load module.

System action: DB2 online refresh of subsystem parameters continues.

DSNZ007I *src* **SUBSYS** *ssnm* **SYSTEM PARAMETERS LOAD MODULE NAME** *module-name* **LOAD COMPLETE**

Explanation: DB2 subsystem *ssnm* has loaded subsystem parameters load module *module-name*, and has notified all resource managers.

DSNZ008I *src* **CSECT** *parm-csect* **NOT FOUND IN SYSTEM PARAMETERS**

Explanation: The named *parm-csect* was not found in the subsystem parameters load module specified in the -SET SYSPARM command. Message DSNZ007I, which was issued prior to this message, identifies the name of the subsystem parameters load module that is missing this CSECT.

System action: DB2 online refresh of subsystem parameters terminates.

System programmer response: Verify that:

- The system parameters load module matches the release level of the DB2 that was started.

© Copyright IBM Corp. 1983, 2008

- The correct load libraries are being used to load the system parameters load module.

If these are correct, then reassemble and relink-edit your system parameters load module.

DSNZ009I *src* **INCORRECT LEVEL** *wrong-level* **FOUND IN** *parm-csect* **SYSTEM PARAMETERS. LEVEL** *correct-level* **IS REQUIRED.**

Explanation: This message, issued during -SET SYSPARM command processing, indicates the specified subsystem parameters load module CSECT *parm-csect* contains an incorrect level identifier. The CSECT contains a level identifier of *wrong-level*.

The active DB2 subsystem requires a level identifier of *correct-level*. This error can occur if the specified subsystem parameters load module does not match the release level of the DB2 system. This error can also occur if the subsystem parameters load module requires additional maintenance. In the latter case, the *parm-csect* is down-level and requires reassembly.

System action: DB2 online refresh of subsystem parameters terminates.

DSNZ010I *src* **SUBSYS** *ssnm* **SYSTEM PARAMETERS LOAD MODULE NAME** *module-name* **NOT FOUND, BLDL R15=***rc***, R0=***rsn*

Explanation: During -SET SYSPARM command processing on subsystem *ssnm*, the specified subsystem parameters load module *module-name* could not be located during MVS BLDL processing. The BLDL issued a return code of *rc* and a reason code of *rsn*.

System action: DB2 online refresh of subsystem parameters terminates.

System programmer response: Verify that the named load module exists, and that DB2 can access it.

DSNZ011I **src SUBSYS ssnm SYSTEM PARAMETERS SET TO STARTUP**

Explanation: DB2 issues this message when the STARTUP parameter is specified on the -SET SYSPARM command. The message indicates that the subsystem parameters specified at DB2 startup are now in effect.

System action: Processing continues.

DSNZ012I *src* **SUBSYS** *ssnm* **SYSTEM PARAMETERS LOAD MODULE NAME** *module-name* **NOT LOADED, LOAD R15=***rc***, R1=***rsn*

Explanation: DB2 issues this message during -SET SYSPARM command processing on subsystem *ssnm*, where the load of *module-name* failed. *rc* and *rsn* are the return and reason codes from the MVS LOAD.

System action: Online refresh of subsystem parameters terminates.

DSNZ013I *src* **SUBSYS** *ssnm* **SYSPARM COMMAND REJECTED, COMMAND ALREADY IN PROGRESS**

Explanation: The current -SET SYSPARM command is rejected due to processing of a previous -SET SYSPARM command.

System action: Online refresh of subsystem parameters for this command is rejected.

User response: Retry the command after the previous -SET SYSPARM command completes.

DSNZ014I *src* **PARAMETER** *parm-name* **IN CSECT** *load-csect-name* **CANNOT BE CHANGED ONLINE. PARAMETER CHANGE IGNORED.**

Explanation: Subsystem parameter *parm-name* in *load-csect-name* is different from its start-up value, but *parm-name* cannot be changed online.

System action: DB2 uses the start-up value for parameter *parm-name*.

DSNZ015I *src* **YOU MUST HAVE INSTALL SYSADM AUTHORITY TO CHANGE PARAMETER** *parm-name* **ONLINE. PARAMETER CHANGE IS IGNORED.**

Explanation: Install SYSADM authority is needed to change the *parm-name* parameter using the SET SYSPARM command. The previous value for *parm-name* is used.

System action: The value for the *parm-name* parameter is not changed in the newly-loaded module.

Chapter 24. DSN1... messages

The messages in this section apply to DSN1LOGP, DSN1COPY, DSN1COMP, DSN1PRNT, and DSN1CHKR.

The value shown for 'severity' is the value returned as the job-stepcondition code from the job step in which the message is issued. If additional messages having higher severity values are issued during the same job step, the higher value is reflected as the job-step condition code.

Severity values have the following meanings:

0 An informational message. No error occurred.

4 A warning message. A condition was detected of which the user should be aware. The user might need to take further action.

8 An error message. An error situation was detected and processing could not continue.

12 A severe error message. A severe error situation was detected and processing could not continue.

DSN1110E LIMIT OF 50 STATEMENTS EXCEEDED

Explanation: DSN1LOGP allows a total of 50 statements for a given job.

Severity: 8 (error)

System action: Processing is terminated.

User response: Resubmit the job using no more than 50 statements.

DSN1111E LIMIT OF 80 TOKENS EXCEEDED

Explanation: DSN1LOGP allows a total of 80 keywords and corresponding value specifications. A keyword with its value is considered two tokens.

Severity: 8 (error)

System action: Processing is terminated.

User response: Resubmit the job using no more than 80 keywords.

DSN1112E TOKEN xx..xx EXCEEDS 64 HEXADECIMAL CHARACTERS

Explanation: The SYSIN input contains the character string 'xx..xx'. This string is invalid.

Severity: 8 (error)

System action: Processing is terminated.

User response: Resubmit the job with the specified token being no more than 64 hexadecimal characters in length.

DSN1113E VALUE FOR KEYWORD xxxxxxxx INVALID

Explanation: The SYSIN input contains the keyword 'xxxxxxxx'. The value specified for that keyword is not of the form '(value)'.

Severity: 8 (error)

System action: Processing is terminated.

User response: Resubmit the job with the correct form of the keyword. The problem might be that the closing parenthesis ')' is missing or that an extra blank exists in the value.

DSN1114E RBASTART MULTIPLY DEFINED

Explanation: The RBASTART parameter can be specified only once for a given job.

Severity: 8 (error)

System action: Processing is terminated.

User response: Resubmit the job using a single RBASTART value.

DSN1115E *csect-name* **KEYWORD** *keyword* **ALLOWS ONLY** *number* **HEX CHARACTERS**

Explanation: The value given for *keyword* is too long.

Severity: 8 (error)

System action: Processing is terminated.

User response: Resubmit the job providing no more than the maximum number of characters allowed for the keyword.

© Copyright IBM Corp. 1983, 2008

DSN1116E *csect-name keyword* **IS USED MORE THAN ONCE.**

Explanation: The *keyword* can be used only once in a single job.

Severity: 8 (error)

System action: Processing is terminated.

User response: Resubmit the job providing a single keyword.

DSN1117E *csect-name keyword* **SPECIFIED WITH NO VALUE OR INCORRECT VALUE**

Explanation: The keyword must be followed by a correct value.

Severity: 8 (error)

System action: Processing is terminated.

User response: Resubmit the job providing a correct value for the keyword.

DSN1118E RBAEND > 12 HEX CHARACTERS

Explanation: The maximum size of a log RBA in DB2 is 6 bytes (hence, 12 characters).

Severity: 8 (error)

System action: Processing is terminated.

User response: Resubmit the job providing an RBAEND value that is no more than 12 hexadecimal characters.

DSN1119E DBID MULTIPLY DEFINED

Explanation: The DBID keyword can be specified only once for a given job.

Severity: 8 (error)

System action: Processing is terminated.

User response: Resubmit the job providing a single DBID value.

DSN1120E DBID SPECIFIED WITH NO VALUE

Explanation: The DBID keyword specifies a hexadecimal database identifier and must be followed by a value.

Severity: 8 (error)

System action: Processing is terminated.

User response: Resubmit the job providing a DBID (value).

DSN1121E DBID > 4 HEX CHARACTERS

Explanation: A DBID is 2 bytes in length (hence, 4 characters maximum).

Severity: 8 (error)

System action: Processing is terminated.

User response: Resubmit the job providing a DBID which is no longer than 4 hexadecimal characters.

DSN1122E OBID MULTIPLY DEFINED

Explanation: The OBID keyword can be specified only once for a given job.

Severity: 8 (error)

System action: Processing is terminated.

User response: Resubmit the job providing a single OBID value.

DSN1123E OBID SPECIFIED WITH NO VALUE

Explanation: The OBID keyword specifies a hexadecimal database object identifier. OBID must be followed by a value.

Severity: 8 (error)

System action: Processing is terminated.

User response: Resubmit the job providing an OBID (value).

DSN1124E OBID > 4 HEX CHARACTERS

Explanation: An OBID is 2 bytes in length (hence, 4 characters maximum).

Severity: 8 (error)

System action: Processing is terminated.

User response: Resubmit the job providing an OBID which is no longer than 4 hexadecimal characters.

DSN1125E PAGE SPECIFIED WITH NO VALUE

Explanation: The PAGE keyword specifies a hexadecimal page number. PAGE must be followed by a value.

Severity: 8 (error)

System action: Processing is terminated.

User response: Resubmit the job providing a PAGE (value).

DSN1126E PAGE > 6 HEX CHARACTERS

Explanation: A page number is 3 bytes in length (hence, 6 characters maximum).

Severity: 8 (error)

System action: Processing is terminated.

User response: Resubmit the job providing a PAGE value which is no longer than 6 hexadecimal characters.

DSN1127E KEYWORD xxxxxxxx UNKNOWN

Explanation: DSN1LOGP does not recognize the keyword 'xxxxxxxx'.

Severity: 8 (error)

System action: Processing is terminated.

User response: Check to make sure all keywords are valid and resubmit the job.

DSN1128E *csect-name keyword* **KEYWORD SPECIFIED WITHOUT** *keyword*

Explanation: The identified keywords are associated with each other. You cannot specify one keyword without the other keyword.

Severity: 8 (error)

System action: Processing is terminated.

User response: Resubmit the job providing the corresponding correct keywords.

DSN1129E PAGE WITHOUT DBID OBID

Explanation: DB2 pages exist in objects called 'page sets'. When you specify a page, you must also identify the page set by using the appropriate DBID and OBID specifications.

Severity: 8 (error)

System action: Processing is terminated.

User response: Resubmit the job providing the associated DBID (value) and OBID (value) that correspond to the given page.

DSN1130E OBID WITHOUT DBID

Explanation: When you specify an OBID, you must also identify the database that contains that OBID by specifying the DBID of the database.

Severity: 8 (error)

System action: Processing is terminated.

User response: Resubmit the job providing both an OBID (value) and a DBID (value).

DSN1131E OFFSET/VALUE PAIR NUMBER xx CONTAINS INVALID HEX DATA

Explanation: A hexadecimal specification is required. The value for offset/value pair number 'xx' contains an invalid hexadecimal character.

Severity: 8 (error)

System action: Processing is terminated.

User response: Resubmit the job providing a correct hexadecimal value specification.

DSN1132E URID SPECIFIED WITH NO VALUE

Explanation: The URID keyword specifies the hexadecimal unit of recovery identifier. URID must be followed by a value.

Severity: 8 (error)

System action: Processing is terminated.

User response: Resubmit the job providing an associated value for the URID parameter.

DSN1133E URID > 12 HEX CHARACTERS

Explanation: The maximum size of a URID in DB2 is 6 bytes (hence, 12 characters).

Severity: 8 (error)

System action: Processing is terminated.

User response: Resubmit the job providing a valid URID value of no more than 12 hexadecimal characters.

DSN1134E *csect-name* **LIMIT OF** *number keyword***S IS EXCEEDED**

Explanation: The *keyword* was specified more than the maximum number of times allowed in any given job.

Severity: 8 (error)

System action: Processing is terminated.

User response: Resubmit the job specifying this keyword no more than the maximum number of times.

DSN1135E SUMMARY MULTIPLY DEFINED

Explanation: The SUMMARY option can be specified only once.

Severity: 8 (error)

System action: Processing is terminated.

User response: Resubmit the job using a singe SUMMARY specification.

DSN1136E SUMMARY SPECIFIED WITH NO VALUE OR INCORRECT VALUE

Explanation: The permitted values for summary are YES, NO, ONLY, or FILTER.

Severity: 8 (error)

System action: Processing is terminated.

User response: Resubmit the job providing a valid parameter for the SUMMARY option.

DSN1137I *csect-name* **FIRST PAGE SET CONTROL RECORD AFTER RESTART LOG** *seq nnnnnnnnnnnn*

Explanation: This message shows the log sequence number of a record that serves as an implicit indication that a restart occurred just prior to this point.

seq RBA or LRSN

nnnnnnnnnnnn
The log sequence number

Severity: 0 (informational)

System action: Processing continues.

DSN1138E *csect-name keyword* **AND** *keyword* **CANNOT BOTH BE SPECIFIED**

Explanation: You cannot specify both keywords in one job.

Severity: 8 (error)

System action: Processing is terminated.

User response: Resubmit the job specifying only one of the keywords.

DSN1139E RID SPECIFIED WITH NO VALUE

Explanation: The RID keyword specifies a 4-byte hexadecimal number, with the first 3 bytes representing the page number and the last byte representing the record ID. A value must be associated with the RID.

Severity: 8 (error)

System action: Processing is terminated.

User response: Resubmit the job providing a RID (value).

DSN1140E RID > 8 HEX CHARACTERS

Explanation: The maximum size of a RID in DB2 is 4 bytes (hence, 8 characters).

Severity: 8 (error)

System action: Processing is terminated.

User response: Resubmit the job providing a valid RID value of no more than 8 hexadecimal characters.

DSN1141E RID WITHOUT DBID OBID

Explanation: The DBID and OBID associated with the given RID value must be specified.

Severity: 8 (error)

System action: Processing is terminated.

User response: Resubmit the job providing the associated DBID (value) and OBID (value) that correspond to the given RID value.

DSN1142E DATAONLY MULTIPLY DEFINED

Explanation: The DATAONLY option limits the log records extracted in the detail report to those that represent data modifications. This option can be specified only once.

Severity: 8 (error)

System action: Processing is terminated.

User response: Resubmit the job specifying DATAONLY once.

DSN1143E DATAONLY SPECIFIED WITH NO VALUE OR INCORRECT VALUE

Explanation: The DATAONLY keyword must be specified with either a YES or NO value. Any other value (including no value) is incorrect.

Severity: 8 (error)

System action: Processing is terminated.

User response: Resubmit the job specifying the correct value for the DATAONLY parameter.

DSN1144E RBASTART SPECIFIED WITH NO VALUE

Explanation: The RBASTART keyword specifies the hexadecimal log RBA from which to begin extraction. RBASTART must be followed by a value.

Severity: 8 (error)

System action: Processing is terminated.

User response: Resubmit the job providing an associated value for the RBASTART parameter.

DSN1146E CURRENT END CHECKPOINT TIMESTAMP TIMESTAMP=yy.ddd hh.mm.ss LOG RBA=xxxxxx IS OUT OF SEQUENCE. THE GREATEST TIMESTAMP SEEN SO FAR IS TIMESTAMP=yy.ddd hh.mm.ss LOG RBA=xxxxxx Multiple processors may be writing to the same log.

Explanation: This message indicates that the current log record has a timestamp that is less than the previous timestamp processed. This may be a potential problem.

Severity: 4 (warning)

System action: Processing continues.

User response: Examine current log to determine whether multiple processors are writing to the same log. (Data may be being overwritten.) This may lead to data inconsistencies.

DSN1147E SYSCOPY MULTIPLY DEFINED

Explanation: The SYSCOPY option can only be specified once.

Severity: 8

System action: Processing is terminated.

User response: Resubmit the job using a single SYSCOPY specification.

System programmer response: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 5.

DSN1148E SYSCOPY SPECIFIED WITH NO VALUE OR INCORRECT VALUE

Explanation: The SYSCOPY keyword must be specified with either a YES or NO value. Any other value (including no value) is incorrect.

This message is issued by the following CSECT: DSN1LPAR

Severity: 8

System action: Processing is terminated.

User response: Resubmit the job specifying the correct value for the SYSCOPY parameter.

System programmer response: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 5.

DSN1150I SUMMARY OF COMPLETED EVENTS

Explanation: This message heads the summary of completed units of recovery (URs) and checkpoints. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Severity: 0 (informational)

System action: Processing continues.

DSN1151I *csect-name mbn* **UR CONNID=***cc* **CORRID=***bb* **AUTHID=***aa* **PLAN=***pp* **START DATE=***yy.ddd* **TIME=***hh:mm:ss* **DISP=***xx* **INFO=***ii* **STARTRBA=***ss* **ENDRBA=***ee* **STARTLRSN=***rr* **ENDLRSN=***nn* **NID=***nid* **LUWID=***luwid* **COORDINATOR=***name* **PARTICIPANTS: (***list of participants***)**

Explanation: This message describes a unit of recovery (UR) that terminated for data sharing member *mbn*.

mbn MEMBER=*membername*, where *membername* is the member of a data sharing group. This is printed only for a data sharing record.

cc Connection ID (for example, BATCH).

bb Correlation ID (for example, JOB xxx).

aa Authorization ID executing the UR.

pp Plan name used by the UR.

DSN1LOGP returns a plan name of SYSTEM in this field if the UR was generated as a result of a system event (such as the first update of a table space). In such a case, no bound plan exists for the UR described.

yy.ddd Starting date of the UR.

hh:mm:ss
Starting time of the UR.

xx Disposition of the UR. Values reported are:
Inflight
In-abort
In-commit
Indoubt
Committed
Aborted
P-Abort

If the UR was bypassed by a conditional restart, DSN1LOGP indicates this by preceding the disposition value with the word BYPASSED.

ii The status of the data is one of the following:
COMPLETE, indicating that all page sets modified by this UR were identified.
PARTIAL, indicating that the list of page sets modified by this UR is incomplete. DSN1LOGP indicates PARTIAL status if all records associated with a UR are not available, and no checkpoint is found before the UR's completion.

ss RBA of the first log record associated with the UR (that is, the URID).

ee RBA of the last log record associated with the UR.

rr LRSN of the first log record associated with the UR.

nn LRSN of the last log record associated with the UR

nid The identifier assigned by IMS or CICS for this UR. This is blank if the UR is a result of TSO, batch, or remote access.

luwid The logical unit of work identifier associated with the UR

name The name of the remote coordinator associated with the UR. This is blank if the coordinator is the local DB2, IMS, or CICS.

list of participants
The names of all remote participants that might have updated data in this UR.

This message is followed by messages that identify the page sets modified by the UR. For each page set, a single message is printed in the following form:

```
DATABASE=dbid=dbname
PAGE SET=obid=psname
```

Where:

dbid Internal database identifier

dbname Database name corresponding to the *dbid*

obid Internal page set identifier

psname Page set name corresponding to the OBID

The *dbname* and *psname* fields might be absent if a full checkpoint was unavailable to DSN1LOGP or was caused by overlapped processing of DB2 during a checkpoint.

If no data was modified by the UR (for example, certain URs might be created merely to track information that is valid only for the life of the application), the phrase NO DATA MODIFIED is printed.

Severity: 0 (informational)

System action: Processing continues.

DSN1152I *csect-name mbn* **UR CONNID=** CORRID=** AUTHID=** PLAN=** START DATE=** TIME=** DISP=***xx*** INFO=***ii*** STARTRBA=***ss*** STARTLRSN=** ENDRBA=***ee*** ENDLRSN=***nn*NID=*nid** LUWID=***luwid*** COORDINATOR=***name*** PARTICIPANTS: (***list of participants***)**

Explanation: This message describes a unit of recovery (UR) for data sharing member *mbn*. The UR identifying information (**) described in message DSN1151I is unavailable.

mbn MEMBER=*membername*, where *membername* is the member of a data sharing group. This is printed only for a data sharing record.

The remaining information (DISP, INFO, START, END, NID, LUWID, COORDINATOR and PARTICIPANTS) appears just as it does in message DSN1151I. If the UR did not complete, the END RBA or LRSN values are not printed.

This message is followed by messages that identify the page sets modified by the UR. Details of that are in the explanation of message DSN1151I.

Severity: 0 (informational)

System action: Processing continues.

DSN1153I *csect-name* **CHECKPOINT** *mbn* **STARTRBA=***ss* **ENDRBA=***ee* **STARTLRSN=***rr* **ENDLRSN=***nn* **DATE=***yy.ddd* **TIME=***hh:mm:ss*

Explanation: This message describes a complete checkpoint on the log starting and ending RBA or LRSN for data sharing member *mbn*. If the information is available, DSN1LOGP also returns the date and time that the checkpoint started.

When this message follows message DSN1157I RESTART SUMMARY, it identifies the checkpoint that would be used at restart. If no checkpoint is available, message DSN1158I is printed instead.

mbn MEMBER=*membername*, where *membername* is the member of a data sharing group. This is printed only for a data sharing record.

Severity: 0 (informational)

System action: Processing continues.

DSN1154I *csect-name* **RESTART** *mbn* **AT RBA=***xx* **LRSN=***ll* **DATE=***yy.ddd* **TIME=***hh:mm:ss*

Explanation: A normal restart occurred at log RBA *xx* or LRSN *ll*.

mbn MEMBER=*membername*, where *membername* is the member of a data sharing group. This is printed only for a data sharing record.

DSN1LOGP also returns the date and time of that restart.

Severity: 0 (informational)

System action: Processing continues.

DSN1155I *csect-name* **CONDITIONAL RESTART** *mbn* **AT RBA=***xx* **LRSN=***ll* **DATE=***yy.ddd* **TIME=***hh:mm:ss* **STARTRBA=***ss* **ENDRBA=***ee* **FORWARD=***ff* **BACKOUT=***bb*

Explanation: A conditional restart occurred at the log

RBA *xx* and LRSN *ll*. DSN1LOGP also returns the date and time of that restart.

mbn MEMBER=*membername*, where *membername* is the member of a data sharing group. This is printed only for a data sharing record.

ss Relative byte address of the beginning of the restart

ee Relative byte address of the conclusion of the restart

ff Whether forward log recovery phase was to be bypassed (YES | NO)

bb Whether backward log recovery phase was to be bypassed (YES | NO)

Severity: 0 (informational)

System action: Processing continues.

DSN1156I ALL URS COMPLETE

Explanation: There are no URs outstanding for restart.

Severity: 0 (informational)

System action: Processing continues.

DSN1157I RESTART SUMMARY

Explanation: This message heads the summary of the description of work to be performed at restart. Restart information that follows is based on the scope of the log scanned. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Severity: 0 (informational)

System action: Processing continues.

DSN1158I NO CHECKPOINT AVAILABLE — RESTART SUMMARY INCOMPLETE

Explanation: No checkpoint is available within the scope of the log scanned. The information following this message includes:

- URs that have not completed
- Page sets modified by these URs
- Page sets with writes pending.

The information cannot be considered complete.

Severity: 0 (informational)

System action: Processing continues.

DSN1159I NO DATABASE WRITES PENDING

Explanation: All database I/O has been completed. If the Log Extractor can find no complete checkpoint, it will return an additional message '(BASED ON INCOMPLETE LOG INFORMATION)', to warn you that there may in fact be pending database writes.

Severity: 0 (informational)

System action: Processing continues.

DSN1160I DATABASE WRITES PENDING:

Explanation: Database writes are pending at restart. Each page set with outstanding I/O is identified by a separate message following this one:

DATABASE=dbid=dbname
PAGESET=obid=psname START=ss

Where:

dbid Internal database identifier

dbname Database name corresponding to the dbid

obid Internal page set identifier

psname Page set name corresponding to the obid

Note: *dbname* and *psname* may be absent if a full checkpoint was unavailable to the Log Extractor or was caused by overlapped processing of DB2 during a checkpoint.

ss Log RBA of the earliest log record required to complete outstanding writes for this page set. If the value of ss is returned as **, this information is unavailable. Such will be the case if the Log Extractor is unable to locate a full checkpoint. In this event, DSN1LOGP will return an additional message (LIST MAY BE INCOMPLETE), to alert you to the possibility that the list of page sets with outstanding writes may be incomplete.

Severity: 0 (informational)

System action: Processing continues.

DSN1162I *csect-name mbn***UR CONNID=***cc* **CORRID=***bb* **AUTHID=***aa* **PLAN=***pp* **START DATE=***yy.ddd* **TIME=***hh:mm:ss* **DISP=***xx* **INFO=***ii* **STARTRBA=***ss* **STARTLRSN=***ll* **NID=***nid***LUWID=***luwid* **COORDINATOR=***name* **PARTICIPANTS: (***list of participants***)**

Explanation: This message describes a unit of recovery (UR) that has not terminated. The information returned is the same as for message DSN1151I, except that the END field is not included in the message because the UR did not terminate.

mbn MEMBER=*membername*, where *membername* is the member of a data sharing group. This is printed only for a data sharing record.

Severity: 0 (informational)

System action: Processing continues.

DSN1163E NO CHECKPOINT AVAILABLE DUE TO LOG ERROR -- RESTART SUMMARY INCOMPLETE

Explanation: A log error was encountered. DSN1LOGP marked any checkpoints encountered before the log error as invalid. There were no complete checkpoints following the log error in the specified log range. The information following this message includes:
- URs that have not completed
- Page set modified by these URs
- Page sets with writes pending.

This information cannot be considered complete.

Severity: 0 (informational)

System action: Processing continues.

DSN1164E INVALID xxxx FOUND AT RBA = yyyyyyyyyyyy

Explanation: While processing the SUMMARY option, an invalid checkpoint record (either URE or RURE identified by 'xxxx') was encountered in the log.

Severity: 0 (informational)

System action: Processing continues.

User response: If the checkpoint record identified in the message is used to restart DB2, the restart will fail because DB2 will not be able to process the unit of recovery presented by the invalid URE or RURE. DB2 must be restarted from a previous checkpoint to correctly process all units of recovery. The print log map (DSNJU004) utility can be used to list the checkpoint records in the BSDS, and the CHKPTRBA parameter of the change log inventory (DSNJU003) utility can be used to direct the DB2 restart to use a particular checkpoint.

DSN1170E SUBTYPE MULTIPLY DEFINED

Explanation: The SUBTYPE parameter can be specified only once for any given job.

Severity: 12 (severe error)

System action: Processing terminates.

User response: Provide a single SUBTYPE value and resubmit the job.

DSN1171E SUBTYPE SPECIFIED WITH NO VALUE

Explanation: The SUBTYPE parameter specifies a decimal set subtype. The SUBTYPE must be followed by a value, i.e. SUBTYPE(value).

Severity: 12 (severe error)

System action: Processing terminates.

User response: Provide a SUBTYPE value and resubmit the job.

DSN1172E SUBTYPE > 2 DECIMAL CHARACTERS

Explanation: The maximum size of a SUBTYPE is 2 decimal characters.

Severity: 12 (severe error)

System action: Processing terminates.

User response: Provide a SUBTYPE value with a maximum of 2 decimal characters and resubmit the job.

DSN1173E *csect-name* **INVALID VALUE** *value* **SPECIFIED FOR KEYWORD** *keyword*

Explanation: The specified keyword value is not valid.

Severity: 12 (severe error)

System action: Processing terminates.

User response: Refer to *DB2 Utility Guide and Reference* for a list of valid values. Provide a valid value and resubmit the job.

DSN1174E OFFSET SPECIFIED WITH NO SUBTYPE

Explanation: The given OFFSET has no corresponding SUBTYPE.

This message is issued by the following CSECT: DSN1LPAR

Severity: 12 (severe error)

System action: Processing terminates.

User response: Provide a valid SUBTYPE for the given OFFSET and resubmit the job.

DSN1175E OFFSET SPECIFIED WITH NO VALUE OR INCORRECT VALUE

Explanation: The given OFFSET has no corresponding VALUE given or an incorrect VALUE is specified, such as specifying another keyword in place of a value.

Severity: 12 (severe error)

System action: Processing terminates.

User response: Provide a valid VALUE for the given OFFSET and resubmit the job.

DSN1176E OFFSET > 8 HEX CHARACTERS

Explanation: The maximum size for an OFFSET is 4 bytes (hence, 8 characters).

Severity: 12 (severe error)

System action: Processing terminates.

User response: Provide an OFFSET with a maximum of 8 characters and resubmit the job.

DSN1177E LIMIT OF 10 OFFSET, VALUE PAIRS EXCEEDED

Explanation: A maximum of 10 OFFSET, VALUE pairs can be specified for a given log record SUBTYPE.

Severity: 12 (severe error)

System action: Processing terminates.

User response: Provide a maximum of 10 OFFSET, VALUE pairs for a given SUBTYPE and resubmit the job.

DSN1178E OFFSET SPECIFIED WITH NO CORRESPONDING VALUE

Explanation: The given OFFSET has no corresponding VALUE keyword specified.

Severity: 12 (severe error)

System action: Processing terminates.

User response: Provide an associated VALUE (value) for the given OFFSET and resubmit the job.

DSN1179E VALUE SPECIFIED WITH NO CORRESPONDING OFFSET OR SUBTYPE PARAMETER

Explanation: The given VALUE has either no corresponding OFFSET or SUBTYPE parameter specified.

Severity: 12 (severe error)

System action: Processing terminates.

User response: Provide an associated OFFSET or SUBTYPE for the given VALUE and resubmit the job.

DSN1180E VALUE SPECIFIED WITH NO VALUE OR INCORRECT VALUE

Explanation: The given VALUE has no corresponding value given, an incorrect value is specified, such as specifying a keyword in place of a value, or an uneven number of characters were specified.

Severity: 12 (severe error)

System action: Processing terminates.

User response: Provide a valid value for the given VALUE and resubmit the job.

DSN1181E VALUE SPECIFIED WITH NO CORRESPONDING OFFSET

Explanation: The given VALUE has no corresponding OFFSET specified.

Severity: 12 (severe error)

System action: Processing terminates.

User response: Provide a valid OFFSET for the given VALUE and resubmit the job.

DSN1182E LIMIT OF 10 LUWIDs EXCEEDED

Explanation: A maximum of 10 LUWID keywords can be specified in any given job.

Severity: 8 (error)

System action: DSN1LOGP processing is terminated.

User response: Resubmit the job providing no more than 10 LUWID keywords.

DSN1183E LUWID SPECIFIED WITH INCORRECT VALUE OR NO VALUE

Explanation: The LUWID consists of an LU network name, an LUW instance number, and a commit sequence number. The LU network name consists of a 1 to 8 character network ID, a period, and a 1 to 8 character network LU name. The LUW instance number consists of a period followed by 12 hexadecimal characters. The LUW sequence number consists of a period followed by 4 hexadecimal characters.

Severity: 8 (error)

System action: DSN1LOGP processing is terminated.

User response: Resubmit the job providing a valid LUWID value.

DSN1184E FILTER KEYWORD MULTIPLY DEFINED

Explanation: The FILTER keyword was specified more than once.

Severity: 8 (error)

System action: DSN1LOGP processing is terminated.

User response: Correct the input to provide only one FILTER keyword and resubmit the job.

DSN1185E FILTER KEYWORD SPECIFIED WITHOUT SUMMARY KEYWORD

Explanation: The FILTER keyword was specified without an associated SUMMARY keyword.

Severity: 8 (error)

System action: DSN1LOGP processing is terminated.

User response: Correct the input to include the SUMMARY keyword and resubmit the job.

DSN1191I DETAIL REPORT OF PAGE REGRESSION ERRORS

Explanation: This message is always generated as the title for a detailed report of page regression errors. It precedes message DSN1192I.

DSN1192I DBID OBID PAGE# CURRENT MEMBER PREV-LEVEL PREV-UPDATE DATE TIME

Explanation: This message contains the title of the detail page regression information.

DSN1193I *dbid obid page# current member prev-level prev_update date time*

Explanation: This message contains the following information about a page regression error:

dbid Hexadecimal DBID of the database that has page regression.

obid Hexadecimal OBID of the pageset that has page regression.

page# Hexadecimal pageset page number that has been regressed.

CURRENT
RBA/LRSN of the current log record at which the page regression has been detected. In non-datasharing, the RBA is printed and in datasharing environment, the LRSN value is printed.

member
The two-byte member ID of the DB2 member that wrote the log record. In non- datasharing, this value is zero.

prev-level
Previous level of the page from the current log record.

prev-update
The previous update to the page found during log processing.

date The date portion of the current log record LRSN value, in printable format.

time The time portion of the current log record LRSN value, in printable format.

DSN1194I SUMMARY REPORT OF PAGE REGRESSION ERRORS

Explanation: This message is always generated as the title for a summary report of page regression errors. It precedes message DSN1195I.

DSN1195I DATABASE SPACENAM DBID OBID #PG-REGRESSIONS

Explanation: This message contains the summary title of the page regression information.

DSN1196I *database spacenam dbid obid #pg-regressions*

Explanation: This message contains the following summary information about page regression errors:

database
The name of the database that has page regression. An asterisk would be printed if the name is unknown.

spacenam
The name of the pageset that has page regression. An asterisk would be printed if the pageset name is unknown.

dbid Hexadecimal DBID of the pageset that has page regression.

obid Hexadecimal OBID of the database that has page regression.

 # pg-regressions
The total number of regressed pages for the pageset.

DSN1197I TOTAL NUMBER OF PAGES CHECKED FOR REGRESSION = *page-count*

Explanation: This message lists the total number of unique pages that DSN1LOGP has checked for page regression errors. *page-count* is displayed in decimal format.

DSN1198I ******NO REGRESSED PAGES DETECTED****

Explanation: No page regression errors were detected within the RBA/LRSN range that was specified.

DSN1210E DSNJSLR ERROR RETCODE=xxxxxxxx REASON CODE =yyyyyyyy

Explanation: The stand-alone log read function detected an error. The return code (RETCODE) is explained in Appendix C (Volume 2) of *DB2 Administration Guide*. The reason code is explained in *DB2 Codes*.

Severity: 8 (error)

System action: Processing is terminated.

DSN1211E DSNJSLR RETURNED BAD LOG RBA

Explanation: One of the three problems listed below exists.

1. The recovery log data set is damaged.
2. You identified a data set that isn't a recovery log data set.
3. The stand-alone log read service has a problem.

Severity: 0 (informational)

System action: Processing continues.

User response: Determine if the problem is your error by dumping the data set and determining if it is a log data set.

Problem determination: A common error is to specify the first data set on an archive tape (the 'Bxxxxxxx' data set) as a log data set; it is actually a BSDS.

DSN1212I *csect-name* **FIRST LOG** *seq* **ENCOUNTERED** *nnnnnnnnnnnn*

Explanation: This message identifies the sequence number of the first log record that was read.

seq RBA or LRSN

Severity: 0 (informational)

System action: Processing continues.

DSN1213I *csect-name* **LAST LOG** *seq* **ENCOUNTERED** *nnnnnnnnnnnn*

Explanation: This message identifies the sequence number of the last log record that was read.

seq RBA or LRSN

Severity: 0 (informational)

System action: Processing continues.

DSN1214I NUMBER OF LOG RECORDS READ nnnnnnnnnnnnnnnn

Explanation: This identifies the number (in decimal) of 'logical' log records read during DSN1LOGP processing.

Severity: 0 (informational)

System action: Processing continues.

DSN1215I NO LOG RECORDS WERE READ

Explanation: DSN1LOGP read no log records.

Severity: 0 (informational)

System action: Processing continues. Possible explanations:

- An error prevents DSN1LOGP from continuing, so no log records have yet been processed. If this is the case, an error message should precede this message.
- You specified the active log data sets or archive log data sets out of RBA sequence.
- You specified a STARTRBA that is greater than any RBA in the active and/or archive data sets available.

DSN1216E *csect-name* **DSNJSLR ERROR RETCODE =** *xxxxxxxx* **REASON CODE =** *yyyyyyyy* *seq* **=** *nnnnnnnnnnnn*

Explanation: The stand-alone log read function returned an error indicating that either the log has an error in one of the control intervals (CI) or a data set containing the requested sequence number cannot be located. The sequence number shown in the message indicates where the error was detected.

seq RBA or LRSN

The sequence number points:

- To the start of the CI, if there is a problem with the log control interval definition (LCID), or with any of the general control information within a CI.
- To the log record in the CI, if there is a problem with a log record header (LRH).

If this is the first log record read during this execution of DSN1LOGP, and if there is a problem with the LCID, the sequence number is all zeros.

Before returning any records, DSN1LOGP checks the control information (LCID) at the end of a CI, and analyzes the LRH to ensure that all records are properly chained together within the CI. If an error is detected while performing this process, DSN1LOGP issues this message before dumping the entire CI. It will not format individual records within the CI, but will, if possible, continue processing by reading the next CI.

Severity: 0 (informational)

System action: Processing continues.

Problem determination: The reason code identifies the nature of the error and is explained in *DB2 Codes*. The return code is explained in Appendix C (Volume 2) of *DB2 Administration Guide* .

DSN1217E *csect-name mbn* **DSNJSLR WARNING RETCODE =** *xxxxxxxx* **REASON CODE =** *yyyyyyyy* **CURRENT LOG** *seq* **=** *nnnnnnnnnnnn*

Explanation: A gap in the log sequence range was encountered for member *mba*. The CURRENT LOG sequence number shows the log record for the member following the gap, and is formatted following this message.

Previous log record can be found with same member ID.

mbn MEMBER=*membername*, where *membername* is the member of a data sharing group. This is printed only for a data sharing record.

seq RBA or LRSN

Severity: 0 (informational)

System action: Processing continues.

Problem determination: The reason code identifies the nature of the error and is explained in *DB2 Codes*. The return code is explained in Appendix C (Volume 2) of *DB2 Administration Guide* .

DSN1218I NUMBER OF LOG ERROR MESSAGES nnnnnnnnnnnnnnn

Explanation: DSN1LOGP distinguishes three classes of errors:

1. Code problems existing in the DB2 or system code used for DSN1LOGP. In the event of a SYSABEND dump, a user abend code of U0153 will be issued, and DSN1LOGP will be terminated.
2. Incorrect invocation of DSN1LOGP caused, perhaps, by your having used an incorrect keyword or missed a DD statement. Under these circumstances, DSN1LOGP will issue appropriate error messages, and the program will be terminated.
3. Failure by a given log CI under the scrutiny of DSN1LOGP. Such scrutiny is performed before any of the records within the CI are processed. Failure is an indication of logical damage, and error messages are issued by the Log Extractor. The CI or failing log record is printed, and DSN1LOGP continues to the next CI or log record.

   The count 'nnnnnnnnnnnnnnn' provided summarizes the number (in decimal) of errors DSN1LOGP detected while accessing the log.

Severity: 0 (informational)

System action: Processing continues.

DSN1219E SYSSUMRY DD STATEMENT MISSING

Explanation: You requested the SUMMARY option, but failed to include the SYSSUMRY DD statement in your JCL invocation.

Severity: 8 (error)

System action: Processing terminates.

User response: Resubmit the job with a SYSSUMRY DD statement included in the JCL invocation.

DSN1220E ARCHIVE LOG TRUNCATED AT xxxxxxxx - INVALID LOG RECORDS WERE READ

Explanation: At a restart of DB2, an archive log was truncated. This archive log data set could not be physically altered to reflect this truncation, and invalid log records therefore still exist. DSN1LOGP has already reported this information in the summary report, and cannot retract it. Nor can it disregard the invalid log information already read in order adequately to summarize what has occurred. Therefore, all information up to this point in the log will be summarized, and a new summary report initiated. Consequently, the same UR may be reported twice with different dispositions and different page sets modified. To avoid this condition, use the BSDS DD statement instead of the ARCHIVE DD statement.

Severity: 0 (informational)

System action: Processing continues.

DSN1221E DSNJSLR ERROR RETCODE=xxxxxxxx REASON CODE=yyyyyyyy VSAM RETURN CODE=aaaa ERROR CODE=bbbb

Explanation: The stand-alone log read function resulted in a VSAM error. The DSNJSLR return code 'xxxxxxxx', and reason code 'yyyyyyyy', included in the message are documented in Appendixes (Volume 2) of *DB2 Administration Guide*.

The VSAM return code (aaaa), and error code (bbbb), identify the nature of the VSAM error. Refer to the appropriate VSAM publication for an explanation of these codes.

Severity: 0 (informational)

System action: Processing continues.

DSN1222E DSNJSLR ERROR RETCODE=xxxxxxxx REASON CODE=yyyyyyyy DYNAMIC ALLOCATION INFORMATION CODE=aaaa ERROR CODE=bbbb

Explanation: An error occurred while dynamically allocating a log data set. DSNJSLR return code 'xxxxxxxx', and reason code 'yyyyyyyy', included in the message, are documented in Appendixes (Volume 2) of *DB2 Administration Guide*. Information code 'aaaa' and error code 'bbbb' were returned by the dynamic allocation SVC and identify the nature of the error. Refer to the appropriate MVS publication for an explanation of these codes.

Severity: 12 (severe error)

System action: Processing terminates.

DSN1223E DSNJSLR ERROR RETCODE=xxxxxxxx REASON CODE=yyyyyyyy RDJFCB RETURN CODE=aaaa

Explanation: An error occurred while trying to read the job file control block. The DSNJSLR return code 'xxxxxxxx', and reason code 'yyyyyyyy', included in the message are documented in Appendixes (Volume 2) of *DB2 Administration Guide* .

The RDJFCB return code (aaaa), identifies the nature of the error. Refer to the appropriate MVS publication for an explanation of these codes.

Severity: 0 (informational)

System action: Processing continues.

DSN1300E *csect-name* **RBA RANGE CANNOT BE USED TO PRINT LOG RECORDS FROM MULTIPLE DB2 SYSTEMS**

Explanation: RBASTART and RBAEND cannot be used to define the range when selecting log records from multiple DB2 systems. LRSNSTART and LRSNEND should be used instead.

System action: Processing terminates.

User response: Resubmit the job with either RBA values only or LRSN values only.

DSN1303E *csect-name* **INCOMPATIBLE DD STATEMENTS WERE SPECIFIED FOR LOG PROCESSING**

Explanation: Either MxxACTn or MxxARCHV was erroneously specified in conjunction with MxxBSDS to locate the log data sets of a member of a DB2 data sharing group for a DSN1LOGP job.

System action: Processing terminates.

User response: Correct the DD statement and resubmit the job.

DSN1309E *csect-name* **SPECIFIED END RANGE IS LESS THAN START RANGE**

Explanation: The value specified for ENDLRSN or ENDRBA is less than the value of STARTLRSN or STARTRBA for a DSN1LOGP job.

System action: Processing terminates.

User response: Resubmit the job with correct start and end values.

DSN1700E VALUE FOR KEYWORD xxxxxxxx IS INVALID

Explanation: The SDMPIN input contains the 'xxxxxxxx'. The value specified for that keyword is invalid.

Severity: 8 (error)

System action: Processing is terminated.

User response: Provide a correct keyword value and resubmit the job.

DSN1701E START TRACE MISSING

Explanation: The START TRACE command is required. The first keyword record of the SDMPIN input does not contain a START TRACE command.

Severity: 8 (error)

System action: Processing is terminated.

User response: Resubmit the job with START TRACE as the first keyword record in the SDMPIN input.

DSN1702E KEYWORD xxxxxxxx MULTIPLY DEFINED

Explanation: Keyword 'xxxxxxxx' was specified multiple times in the SDMPIN input. All keywords are allowed only once.

Severity: 8 (error)

System action: Processing is terminated.

User response: Specify the keyword only once. Resubmit the job.

DSN1703E KEYWORD xxxxxxxx UNKNOWN

Explanation: DSN1SDMP does not recognize the keyword 'xxxxxxxx'.

Severity: 8 (error)

System action: Processing is terminated.

User response: Make sure that all keywords are valid and then resubmit the job.

DSN1704E UNABLE TO OBTAIN RETURN-AREA STORAGE

Explanation: An error resulted when DSN1SDMP attempted to obtain storage for an instrumentation facility interface (IFI) return area.

Severity: 8 (error)

System action: Processing is terminated.

User response: When you resubmit the job either:
- Specify a larger region size in the job control language.
- Reduce the value of BUFSIZE on START TRACE.

DSN1705E SYNTAX ERROR, LINE = ll, COLUMN = cc

Explanation: A syntax error was encountered in the SDMPIN input. The location of the error is given by:

ll The line number within the input stream that is in error.

cc The column number.

Severity: 8 (error)

System action: Processing is terminated.

User response: Correct the syntax error and then resubmit the error.

DSN1706E INVALID ABEND CODE SPECIFIED

Explanation: An invalid abend code was specified on the ACTION keyword.

Severity: 8 (error)

System action: Processing is terminated.

User response: Allowable abend codes are 00E60100 through 00E60199. Specify an allowable abend code and then resubmit the job.

DSN1707E LIMIT OF 50 STATEMENTS EXCEEDED

Explanation: DSN1SDMP allows a maximum of 50 statements for a given job.

Severity: 8 (error)

System action: Processing is terminated.

User response: Use no more than 50 statements when you resubmit the job.

DSN1708E TOO MANY SELECT CRITERIA VALUES

Explanation: DSN1SDMP allows a maximum of eight SELECT criteria values.

Severity: 8 (error)

System action: Processing is terminated.

User response: Use no more than eight SELECT criteria values and then resubmit the job.

DSN1709E INPUT/OUTPUT ERROR, DDNAME=xxxxxxxx

Explanation: An error resulted when DSN1SDMP attempted to do I/O. The ddname of the data set to which I/O failed is 'xxxxxxxx'

Severity: 8 (error)

System action: Processing is terminated.

User response: Ensure that the 'xxxxxxxx' DD statement in the job control language is valid and resubmit the job.

DSN1710E UNABLE TO OPEN SDMPTRAC

Explanation: An error resulted when DSN1SDMP attempted to open the SDMPTRAC data set.

Severity: 8 (error)

System action: Processing is terminated.

User response: Ensure that the SDMPTRAC DD statement in the job control language is valid and resubmit the job.

DSN1711E START TRACE ERROR, REASON=xxxxxxxx

Explanation: DB2 did not process the submitted START TRACE command. The reason code 'xxxxxxxx' is returned from the instrumentation facility interface (IFI).

Severity: 8 (error)

System action: Processing is terminated.

User response: Check the return area that follows message DSN1712I. Possible problems are:

- The syntax of the START TRACE command is invalid. Correct the command and resubmit the job. Do not include the subsystem recognition character before the command text.
- The user does not have authority to issue trace commands. Obtain trace authority and resubmit the job.

DSN1712I START TRACE RETURN-AREA FOLLOWS:

Explanation: DSN1SDMP issued the START TRACE command and the return area resulting from the instrumentation facility interface (IFI) COMMAND call follows this message. The return area contains DB2 command response messages.

Severity: 0 (informational)

System action: Processing continues.

DSN1713I READA ERROR, REASON=xxxxxxxx

Explanation: An error resulted when DSN1SDMP issued a instrumentation facility interface (IFI) READA call in an attempt to obtain DB2 trace records. Reason code 'xxxxxxxx' is the IFI reason code returned in IFCARC2.

Severity: 0 (informational)

System action: DSN1SDMP writes any records that were returned and processing continues.

User response: Possible reason codes are:

00E60802
The DSN1SDMP return area was not large enough to hold the returned trace records. This condition should not occur because the DSN1SDMP return area is the same size as the requested OPn monitor buffer.

00E60803
No trace data was available when the READA request was made. This condition occurs when the trace started by DSN1SDMP is stopped from the operator console or by another application program.

DSN1714I TRACE RECORDS WERE LOST

Explanation: DB2 trace records were lost and are not contained in the SDMPTRAC data set.

Severity: 0 (informational)

System action: Processing continues.

User response: Possible reasons for lost trace records are:

- The OP buffer is not large enough. Specify a larger BUFSIZE on the START TRACE command and then resubmit the job.
- The trace data is too voluminous and buffer is overflowing before the data can be read. Specify a more granular selectivity on the START TRACE command and then resubmit the job.

DSN1715E UNABLE TO OBTAIN WORK SPACE

Explanation: An error resulted when DSN1SDMP attempted to obtain work area storage.

Severity: 8 (error)

System action: Processing is terminated.

User response: Specify a larger region size in the job control language and then resubmit the job.

DSN1716E UNABLE TO OPEN SDMPIN

Explanation: An error resulted when DSN1SDMP attempted to open the SDMPIN data set.

Severity: 8 (error)

System action: Processing is terminated.

User response: Ensure that the SDMPIN DD statement in the job control language is valid and resubmit the job.

DSN1717I STOP TRACE RETURN-AREA FOLLOWS:

Explanation: DSN1SDMP issued the STOP TRACE command and the return area resulting from the instrumentation facility interface (IFI) COMMAND call follows this message. The return area contains DB2 command response messages.

Severity: 0 (informational)

System action: Processing continues.

DSN1718E STOP TRACE ERROR, REASON=xxxxxxxx

Explanation: DSN1SDMP attempted to submit a DB2 STOP TRACE command at the end of its processing; however, the command failed. The reason code returned from the instrumentation facility interface (IFI) is 'xxxxxxxx'.

Severity: 8 (error)

System action: Processing is terminated.

User response: Check the return area that follows message DSN1717I. One possible problem is:

- The trace has already been stopped from the operator's console or by another application program.

DSN1719I ACTIONS ARE COMPLETE

Explanation: All requested IFC selective dump actions finished. All trace records have been written to the specified destination(s). If an abend action was requested, all abends have taken place.

Severity: 0 (informational)

System action: DSN1SDMP stops the trace that it started and completes processing.

DSN1720I READA ERROR, REASON=xxxxxxxx

Explanation: An error resulted when DSN1SDMP issued a instrumentation facility interface (IFI) READA call in an attempt to obtain DB2 trace records. Reason code 'xxxxxxxx' is the IFI reason code returned in IFCARC2.

Severity: 8 (error)

System action: Processing is terminated.

User response: One possible reason code is:

00E60824
The user does not have correct monitor authority. Obtain the correct monitor authority (MONITOR1 or MONITOR2) and resubmit the job.

DSN1800I START OF DSN1CHKR FOR JOB job-name step-name

Explanation: This message indicates that DSN1CHKR started processing.

Severity: 0 (informational)

System action: DSN1CHKR starts processing.

DSN1801I ANCHOR POINT mm IN ERROR IN PAGE nnnnnn

Explanation: This message indicates that anchor point 'mm' is in error in page 'nnnnnn'. Either the backward pointer is not zero when the forward pointer is zero, or the backward pointer does not point to the last record ID in the parent chain. If this message is preceded by DSN1835I then the child pointer is pointing to a page beyond the high used RBA of the data set. Either the child pointer is invalid or the data set has been truncated.

Severity: 4 (warning)

System action: The page is marked for output if the FORMAT/DUMP option is in effect. Processing continues.

Problem determination: Contact your IBM Support Center for assistance.

DSN1802I RING/LINK CHAIN BROKEN - type RID = nnnnnnnn

Explanation: This message indicates that there is a broken ring or link chain beginning at 'type' (anchor or parent) with record id equal to 'nnnnnnnn'. This is one of a series of messages associated with a broken ring or link. It will appear in combination with DSN1803I, DSN1804I, and DSN1805I.

Severity: 4 (warning)

System action: The page is marked for output if the FORMAT/DUMP option is in effect. Processing continues.

Problem determination: Contact your IBM Support Center for assistance.

DSN1803I PARENT RID = nnnnnnnn

Explanation: This message supplies the record id 'nnnnnnnn' for the parent record of a detected broken ring or link. This is one of a series of messages associated with a broken ring or link. It will appear in combination with DSN1802I, DSN1804I, and DSN1805I.

Severity: 4 (warning)

System action: Processing continues.

Problem determination: Contact your IBM Support Center for assistance.

DSN1804I CURRENT RID = nnnnnnnn

Explanation: This message supplies the record id 'nnnnnnnn' for the current record at the time the broken ring or link was detected. This is one of a series of messages associated with a broken ring or link. It will appear in combination with DSN1802I, DSN1803I, and DSN1805I.

Severity: 4 (warning)

System action: Processing continues.

Problem determination: Contact your IBM Support Center for assistance.

DSN1805I PREVIOUS IN CHAIN = nnnnnnnn

Explanation: This message supplies the record id 'nnnnnnnn' of the record prior to when the broken ring or link was detected. This is one of a series of messages associated with a broken ring or link. It will appear in combination with DSN1802I, DSN1803I, and DSN1804I.

Severity: 4 (warning)

System action: Processing continues.

Problem determination: Contact Your IBM Support Center for assistance.

DSN1806I IDMAP ENTRY entry-number IN PAGE page-number IS IN ERROR

Explanation: This message indicates that the IDMAP entry 'entry-number' in page 'page-number' is not valid. Valid entries are from X'0014' to X'0Fhh' (depending on the number of IDMAP entries in the page).

Severity: 4 (warning)

System action: The page is marked for output if the FORMAT/DUMP option is in effect. Processing continues.

Problem determination: Contact your IBM Support Center for assistance.

DSN1807I CHILD POINTER chd-rid IN ERROR IN PARENT RECORD par-rid IN PAGE pppppp

Explanation: This message indicates that the parent record with record ID 'par-rid' in page 'pppppp' has a child pointer with an invalid record ID of 'chd-rid'. If this message is preceded by DSN1835I, then the child pointer is pointing to a page beyond the high used RBA of the data set. Either the child pointer is invalid or the data set has been truncated.

Severity: 4 (warning)

System action: The page is marked for output if the FORMAT/DUMP option is in effect. Processing continues.

Problem determination: Contact your IBM Support Center for assistance.

DSN1808I OVERFLOW RECORD IN ERROR - OVERFLOW RID = ovf-rid, POINTER RID = ptr-rid

Explanation: This message indicates that the record at record id 'ovf-rid', pointed to by the pointer record at record id 'ptr-rid', is not an overflow record.

Severity: 4 (warning)

System action: The page is marked for output if the FORMAT/DUMP option is in effect. Processing continues.

Problem determination: Contact your IBM Support Center for assistance.

DSN1809I OVERFLOW RECORD IN ERROR - OVERFLOW RID = ovf-rid, HAS NO POINTER RECORD

Explanation: This message indicates no pointer record points to the overflow record with record id 'ovf-rid'.

Severity: 4 (warning)

System action: The page is marked for output if the FORMAT/DUMP option is in effect. Processing continues.

Problem determination: Contact your IBM Support Center for assistance.

DSN1810I INPUT DATA SET INVALID, NOT A CHECKABLE SYSTEM TABLESPACE

Explanation: This message indicates that the data set defined by SYSUT1 is not a checkable table space and therefore cannot be analyzed.

Severity: 8 (error)

System action: Processing terminates.

System programmer response: Correct the SYSUT1 DD statement and rerun the job.

Problem determination: Checkable system table spaces must be referenced by a hashed entry or contain links. Single-record table spaces referenced by an index are not checkable. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSN1811I ANCHOR POINT PAGE page IS OUT OF RANGE — VALUE IS LESS THAN 2 OR GREATER THAN 'E'X

Explanation: The ANCHOR function has an incorrect page value for the page-slot argument. Anchor points can be found only on pages 2 through 14 (E in hexadecimal). Any anchor point page specified beyond that range is invalid.

Severity: 4 (warning)

System action: Processing continues with the next set of parameters.

DSN1812I ORPHAN ID = orph-id ID ENTRY = entry-val FOUND IN PAGE page-number

Explanation: This message indicates that the record with id = 'orph-id' and id entry = 'entry-val' in page 'page-number' is an orphan.

Severity: 4 (warning)

System action: Processing continues.

Problem determination: Contact your IBM Support Center for assistance.

DSN1813I UNABLE TO DUMP PAGE page-number

Explanation: An error was encountered while retrieving the page and therefore page 'page-number' could not be dumped.

Severity: 4 (warning)

System action: Processing continues.

Problem determination: Examine previous diagnostic and system messages, correct problem, and rerun job.

DSN1814I UNEXPECTED END OF DATA AT PAGE page-number

Explanation: The END-OF-DATA condition was reached before expected at page 'page-number'.

Severity: 8 (error)

System action: Processing terminates.

Problem determination: Examine previous diagnostic and system messages, correct problem and rerun job.

DSN1815I DSN1CHKR COMPLETED SUCCESSFULLY, pg-cnt PAGES PROCESSED

Explanation: This message indicates the DSN1CHKR successfully completed processing; all checks completed. The 'pg-cnt' represents the number of pages read during check processing, not the number of pages in the table space.

Severity: 0 (informational)

System action: Processing terminates.

DSN1816I DSN1CHKR TERMINATED WITH ERRORS, pg-cnt PAGES PROCESSED

Explanation: This message indicates the DSN1CHKR did not successfully complete processing. The 'pg-cnt' represents the number of pages read during check processing, not the number of pages in the table space.

Severity: 4 (warning) or 8 (error)

System action: Processing terminates.

Problem determination: Examine previous diagnostic messages, correct problems if possible and rerun job. Otherwise contact your IBM Support Center for assistance.

DSN1817I ARGUMENT arg-val OF THE MAP COMMAND IS INVALID

Explanation: This message indicates that the argument 'arg-val' of the MAP command is invalid.

Severity: 8 (error)

System action: Processing terminates.

Problem determination: Refer to *DB2 Utility Guide and Reference* for the correct syntax.

Examples of invalid arguments are:

- The offset value of the ANCHOR or HASH option is neither zero (0) nor four (4).
- When the ANCHOR and RID option has been specified, the page number specified is less than two, which would be a nondata page.

DSN1818I IDMAP ENTRY entry-number IN PAGE page-number IS BEING USED BY MULTIPLE CHAINS - ONE OF WHICH IS IN ERROR

Explanation: This message indicates that a record is part of two different chains and therefore creates an invalid structure within the table space. If using the MAP option, the mapping of the chain stops since the path back to the starting record is lost.

Severity: 4 (warning)

System action: Processing continues.

Problem determination: Contact your IBM Support Center for assistance.

DSN1819I HASH VALUE FOR dbid CORRESPONDS TO: PAGE = page-number AND ANCHOR POINT OFFSET = offset

Explanation: This message indicates the hash values for the given 'dbid', after applying the hash function, where 'page-number' is the page and 'offset' is the anchor point.

Severity: 0 (informational)

System action: Processing continues.

DSN1820I MAPPING OF map-opt (val1,val2) HAS STARTED

Explanation: This message indicates that mapping of the indicated option 'map-opt', which is either ANCHOR, HASH, PAGE or RID, has started. The option's arguments, 'val1' and 'val2' will also be mapped. Following this message will be the records of the chain.

Severity: 0 (informational)

System action: Processing continues.

DSN1821I MAPPING OF 'map-opt' (val1,val2) HAS COMPLETED

Explanation: This message indicates that mapping of the indicated option 'map-opt', which is either ANCHOR, HASH, PAGE, or RID, started. The option's arguments, 'val1' and 'val2', will also be mapped. Preceding this message will be the records of the chain.

Severity: 0 (informational)

System action: Processing continues.

DSN1822I DBID dbid HASHED TO AN EMPTY ANCHOR POINT - NO ENTRIES FOR THE GIVEN DBID ARE AVAILABLE IN THE TABLESPACE

Explanation: DBID 'dbid' hashed to an empty anchor point. An anchor point with a value of 0 (empty) means that no entries are available in the table space. Therefore, no links are checked.

Severity: 4 (warning)

System action: Processing continues with the next set of parameters.

DSN1823I INVALID SYNTAX, c EXPECTED

Explanation: This message indicates that an unexpected character was found while parsing the parmlist and that 'c' was the expected character.

Severity: 8 (error)

System action: Processing terminates.

Problem determination: Refer to *DB2 Command Reference* for the correct syntax.

DSN1824I INVALID HEX DATA: hex-data

Explanation: This message indicates that the input value 'hex-data' is not a valid hexadecimal number.

Severity: 8 (error)

System action: Processing terminates.

Problem determination: Correct input value and rerun job.

DSN1825I INPUT VALUE EITHER TOO LONG OR OMITTED: in-val

Explanation: This message indicates that the input value 'in-val' is either greater than 8 hexadecimal digits or was totally omitted.

Severity: 8 (error)

System action: Processing terminates.

Problem determination: Correct input value and rerun job.

DSN1826I RECORD WITH OBID = rec-obid IS INVALID FOR THIS CHAIN

Explanation: This message indicates that the record with OBID 'rec-obid' is invalid for the chain currently being processed, assuming the OBID of the parent or first record processed was not in error.

Severity: 4 (warning)

System action: Processing terminates.

Problem determination: If using the MAP function, make sure the given values point to a valid record. Otherwise, contact your IBM Support Center for assistance.

DSN1827I THE SPECIFIED PAGE IS NOT A HASHED DATA PAGE - THE REQUESTED FUNCTION REQUIRES A HASHED DATA PAGE.

Explanation: This message indicates that the HASH or ANCHOR option of the MAP command was issued against a nonhashed data page (that is, a page with no anchor points).

Severity: 4 (warning)

System action: Processing continues with the next set of parameters.

Problem determination: If using the ANCHOR function, examine the page-slot argument to ensure that the page value is correct. If using the HASH function, ensure that the 'dbid' value is correct. Otherwise, examine the SYSUT1 DD statement to ensure that the table space is correct. If so, use the RID or PAGE function to achieve the desired mapping.

DSN1828I OBID = rec-obid IS INVALID FOR THIS DATABASE

Explanation: This message indicates that the record with OBID 'rec-obid' is invalid for the database currently being processed.

Severity: 4 (warning)

System action: Processing continues with the next set of parameters.

Problem determination: If using the MAP command, insure that all user supplied offsets actually point to valid data. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSN1829I MAPPING MUST STOP AT PARENT RECORD WHEN USING THE BACKWARD POINTER

Explanation: This message indicates that the requested mapping stopped at the parent record rather than the starting record because of the use of backward pointers.

Severity: 4 (warning)

System action: Processing continues.

DSN1830I INVALID NUMBERS OF VALUES SPECIFIED FOR func-name

Explanation: This message indicates that either:

- No value was specified.
- An odd number of values was specified for a function that requires an even number of values.

Severity: 8 (error)

System action: Processing terminates.

Problem determination: Refer to *DB2 Utility Guide and Reference* for the parameter specifications. Correct the parmlist and rerun the job.

DSN1831I END OF DATE REACHED, PAGE = page-number IS OUT OF RANGE

Explanation: This message indicates that page 'page-no', specified as a parameter of the MAP option, caused the END-OF-DATA condition to be reached. The page number is invalid for the table space being mapped.

Severity: 4 (warning)

System action: Processing continues with the next set of parameters.

Problem determination: Correct the page parameter of the MAP option and rerun the job.

DSN1833I ANCHOR POINT IS NOT IN USE

Explanation: The anchor point to be mapped has a value of 0, which means that it is not in use. Therefore, no links are checked.

Severity: 4 (warning)

System action: Processing continues with the next set of parameters.

DSN1834I GETMAIN FOR BUFFERS FAILED. NO MORE AVAILABLE STORAGE IN ADDRESS SPACE.

Explanation: The GETMAIN for page buffers failed. There is no more virtual storage available in the address space. The table space being checked is too large to be processed with the current system configuration.

Severity: 8 (error)

System action: Processing terminates.

Problem determination: Make changes to the system configuration, if possible, to allow more virtual storage to be available in the address space.

DSN1835I THE REQUESTED PAGE page-no IS BEYOND THE END OF THE TABLESPACE.

Explanation: A request for a page number from an anchor point or a child pointer in a parent record is pointing beyond the high used RBA of the data set. Either the rid is in error or the data set has been truncated.

Severity: 4 (warning)

System action: The page is marked for output if the FORMAT/DUMP option is in effect. Processing continues.

Problem determination: Examine previous diagnostic and system messages, correct problem if possible and rerun the job. Otherwise, contact your IBM Support Center for assistance.

DSN1930I *csect-name* **THE INPUT PARAMETER** *xxxxxxxx* **CONFLICTS WITH THE HEADER PAGE FIELD HPG***yyyy***. THE INPUT PARAMETER WILL BE USED**

Explanation: The specified parameter does not agree with the value of the related field in the header page. The utility will continue the execution using the specified input parameter.

Severity: 0 (informational)

System action: The utility continues

User response: Confirm that the input parameter has the correct value.

DSN1931I *csect-name* **THE INPUT PARAMETER** *xxxxxxxx* **CONFLICTS WITH THE HEADER PAGE FIELD HPG***yyyy***. THE HEADER PAGE VALUE WILL BE USED**

Explanation: The specified parameter does not agree with the value of the related field in the header page. The utility will continue the execution using the header page value.

Severity: 0 (informational)

System action: The utility continues

User response: Confirm that the input parameter has the correct value.

DSN1939I *csect-name* **AN INVALID CI SIZE HAS BEEN DETECTED**

Explanation: The VSAM Control Interval (CI) size for the output data set is invalid. The CI size must be 4K, or it must match the page size.

Severity: 8 (error)

System action: Processing terminates.

System programmer response: Redefine the output VSAM data set so that the CI size is either 4K or equal to the page size of the object that is being processed.

DSN1940I *csect-name* **COMPRESSION REPORT**

noncmpbytes
KB WITHOUT COMPRESSION

cmpbytes
KB WITH COMPRESSION

percsave
PERCENT OF THE BYTES WOULD BE SAVED

numdictrows
ROWS SCANNED TO BUILD DICTIONARY

numscanrows
ROWS SCANNED TO PROVIDE COMPRESSION ESTIMATE

numdictents
DICTIONARY ENTRIES

auncmprowlen
BYTES FOR AVERAGE UNCOMPRESSED ROW LENGTH

avgcmprowlen
BYTES FOR AVERAGE COMPRESSED ROW LENGTH

dictpages
DICTIONARY PAGES REQUIRED

noncmppages
PAGES REQUIRED WITHOUT COMPRESSION

cmppages
PAGES REQUIRED WITH COMPRESSION

pctpgsave
PERCENT OF THE DB2 DATA PAGES WOULD BE SAVED

Explanation: DSN1COMP completed successfully. This report contains the compression savings estimate based on the input data set. For a sample of the report, see the DSN1COMP utility in *DB2 Utility Guide and Reference*.

The compression statistics follow:

noncmpbytes
The total number of uncompressed KB (where

KB = 1024 bytes) for the data rows (including the row prefix). This number was rounded to the nearest KB.

cmpbytes
The total number of compressed KB for the data rows (including the row prefix). This number was rounded to the nearest KB.

percsave
The percentage of bytes that would be saved if the rows were compressed.

The values above reflect only the number of bytes for the data row and the associated row prefix (6 bytes). DB2 page and dictionary page overhead is not included.

numdictrows
The number of data rows that were used to build the dictionary.

numscanrows
The number of data rows that were processed using the dictionary to produce the compression savings estimate.

numdictents
The number of entries contained in the compression dictionary.

auncmprowlen
Average uncompressed row length.

avgcmprowlen
Average compressed row length.

dictpages
The number of DB2 pages required for the compression dictionary.

noncmppages
The estimated number of DB2 pages required by the uncompressed data rows. Space required for PCTFREE and FREEPAGE is included. This number was rounded up to the next integer.

cmppages
The estimated number of DB2 pages required by the compressed data rows. Space required for PCTFREE and FREEPAGE is included. Dictionary pages are also included. This number was rounded up to the next integer.

pctpgsave
An estimate of the percentage of DB2 pages that would be saved by compressing the data rows.

Severity: 0 (informational)

System action: The utility completed successfully.

DSN1941I *csect-name* **COMPRESSION ESTIMATE WAS NOT CALCULATED**

Explanation: There is insufficient data to build a compression dictionary. Either the input data set is empty, or the number of rows in the input data set was insufficient to build the dictionary.

Severity: 8 (error)

System action: Processing terminated.

User response: Increase the number of rows in the input data set.

DSN1942I *csect-name* **ERROR WHEN BUILDING THE COMPRESSION DICTIONARY**

Explanation: An internal error was encountered while building the dictionary.

Severity: 8 (error)

System action: Processing terminated.

System programmer response: Determine the failing environment. Refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSN1943I *csect-name* **DICTIONARY PAGE ENCOUNTERED**

Explanation: A dictionary page was encountered while processing the input data set. This indicates that the data is in compressed format.

Severity: 8 (error)

System action: Processing terminated.

User response: Run DSN1COMP on a data set containing data that is not in a compressed format.

DSN1944I *csect-name* **INPUT PARAMETERS**

Explanation: The input parameters used by DSN1COMP are either user-supplied or default values.

Severity: 0 (informational)

System action: Processing continues.

DSN1945I *csect-name* **DICTIONARY TREE**

Explanation: The compression dictionary tree follows. Each node is formatted in the SYSPRINT data set.

Severity: 0 (informational)

System action: Processing continues.

DSN1946I *csect-name* **IMAGE COPY CONTAINING MULTIPLE PARTITIONS**

Explanation: DSN1COMP will build a compression report given sufficient data in the first partition only, of an Image Copy containing multiple partitions.

Severity: 8 (error)

System action: Processing terminated.

User response: Run DSN1COMP on an image copy containing a single partition.

DSN1947I **DSN1947I** *csect-name* **COMPRESSION UNAVAILABLE FOR INDEX SPACES**

Explanation: Index data was provided as input to DSN1COMP. Compression is supported only for table space data.

Severity: 8 (error)

System action: Processing terminated.

User response: Provide DSN1COMP with data containing table space data.

DSN1950I **DSN1COPY HAS SUPPRESSED VSAM OPEN VERIFY, ACBERRFLG=acb-error-code.open-option**

Explanation: DSN1COPY suppresses the VSAM VERIFY done by the VSAM OPEN for the SYSUT1 data set. The VSAM VERIFY is not needed for the VSAM data sets used by DB2. For DB2 VSAM data sets, the data set hi-used RBA in the ICF catalog is always current.

This message is issued by the following CSECT: DSN1COPY

Severity: 0 (informational)

System action: Processing continues.

DSN1951I **ODD NUMBER OF HEX CHARACTERS SPECIFIED IN VALUE PARM.**

Explanation: The VALUE parameter contained an odd number of hexadecimal characters. Only even numbers of hexadecimal characters are allowed.

This message is issued by the following CSECT: DSN1VAL

Severity: 8 (error)

System action: Processing terminates.

User response: Correct the number of hexadecimal digits specified for the VALUE parameter and resubmit the job.

Problem determination: Refer to DSN1COPY documentation for parameter requirements in *DB2 Utility Guide and Reference* .

DSN1952I **UNEXPECTED END OF INPUT DATA - CHECK FOR INVALID SPECIFICATION OF THE 32K PARAMETER.**

Explanation: End of data occurred for the image copy data set specified in SYSUT1 before the first page could be read. Either the wrong input data set was specified or 32KB was specified even though the input data set contains 4KB pages.

This message is issued by the following CSECT: DSN1COPY

Severity: 8 (error)

System action: Processing terminates.

User response: Remove the 32KB parameter if the input data set contains 4KB pages, change the name of the input data set to specify a valid image copy, or remove the FULLCOPY or INCRCOPY parameter.

Problem determination: Refer to DSN1COPY documentation for parameter requirements.

DSN1953I **STARTING PAGE NUMBER REQUIRES AN INPUT DATA SET NAME OF data-set-name**

Explanation: The starting page number of the page range to be printed is inconsistent with the input data set specified.

Severity: 8 (error)

System action: Processing terminates.

User response: Ensure that NUMPARTS, the 32KB parameter (or its default of 4KB), and the starting page number are correct. If it is correct, change the input data set name as indicated by the message.

Problem determination: Refer to DSN1COPY or DSN1PRNT documentation for parameter requirements.

DSN1954I **csect-name PARAMETER PAGESIZE OR LOB IS MISSING OR INCORRECTLY SPECIFIED.**

Explanation: This message is issued in the following situations:

- The program tried to determine the page size from the first page of the data set, but the first page is not a header page, or
- The first page is a LOB header page and the LOB parameter is missing.

Severity: 8 (error)

System action: Processing terminates.

User response: If the input data set is not a LOB table

space, specify the PAGESIZE parameter. If the input data set is a LOB table space, specify the LOB or PAGESIZE parameter or both, as appropriate.

DSN1955I SYSUT1 DD CARD IS MISSING

Explanation: No SYSUT1 DD statement was provided.

Severity: 8 (error)

System action: Processing terminates.

User response: Rerun the job with a SYSUT1 DD statement.

DSN1956I *csect-name* **SYNTAX ERROR:** *parm2* **IS SPECIFIED WITHOUT** *parm1*

Explanation: The keyword *parm1* must be specified with *parm2*.

Severity: 8 (error)

System action: Processing terminates.

User response: Resubmit the job specifying the correct keyword.

DSN1957I *csect-name* **COMPRESSED RECORDS CANNOT BE EXPANDED DUE TO INCORRECT INPUT DATA SET SPECIFICATION.**

Explanation: DSN1PRNT issues this message when
the EXPAND option is specified. The input data set
might be an incremental image copy data set or an
inline image copy data set.

Severity: 8 (error)

System action: Processing terminates.

User response: Resubmit the job specifying the correct input data set.

DSN1958I INVALID DATA SET NAME.

Explanation: A DB2 table space was given an incorrect name.

Severity: 8 (error)

System action: Processing terminates.

User response: Correct the output data set name if FULLCOPY or INCRCOPY were specified. If a page range is being printed, correct the input data set name. Refer to Part 2 (Volume 1) of *DB2 Administration Guide* for a discussion of DB2 data set naming conventions.

DSN1959I BOTH FULLCOPY AND INCRCOPY PARAMETERS CANNOT BE SPECIFIED.

Explanation: DSN1COPY cannot apply to both full images copies and incremental image copies in the same execution. The full image copies are applied first. Each incremental image copy must then be applied in separate steps, starting with the oldest incremental copy.

This message is issued by the following CSECT: DSN1SDEF

Severity: 8 (error)

System action: Processing will terminate.

User response: Omit either FULLCOPY or INCRCOPY.

DSN1960I *csect-name input/output* **DATA SET ALLOCATION ERROR. ERROR FIELD IS ERRFIELD** *value1***. INFORMATION FIELD IS INFFIELD** *value2***.**

Explanation: The allocation of an input or output data set failed. The ERRFIELD *value1* contains the error number; the INFFIELD *value2* contains the information number. See SVC99 error codes in *z/OS MVS Diagnosis: Reference* or *z/OS MVS Programming: Authorized Assembler Services Guide* for help in determining the cause of the error.

Severity: 8 (error)

System action: Processing terminates.

User response: Correct the cause of the allocation error and resubmit the job.

DSN1961I PIECE NUMBER value IS INVALID.

Explanation: A page was retrieved that has a piece number greater than NUMPARTS for partitioned table spaces or greater than the default of 32 for linear table spaces. 'DPIECNUM value' is the piece number for the retrieved page. For page range printing, this message is issued if the starting page number has an invalid piece number as determined from calculations involving NUMPARTS and the page size.

Severity: 8 (error)

System action: Processing will terminate.

User response: Look in SYSIBM.SYSTABLESPACE for the correct number of partitions for this page. Correct NUMPARTS or 32KB if they were specified incorrectly, resubmit the job with a valid image copy, or correct the starting page number for page range printing.

DSN1962I OUTPUT DATA SETS MUST BE VSAM WHEN IMAGE COPY DATA SETS ARE USED AS INPUT TO DSN1COPY.

Explanation: If image copy data sets are used as input to DSN1COPY, the output data sets must be VSAM. QSAM data sets cannot be used for output.

Severity: 8 (error)

System action: Processing will terminate.

User response: Change the output data set to VSAM, or remove the FULLCOPY and NUMPARTS parameters.

DSN1963I ZERO PAGES ARE INVALID WHEN IMAGE COPY DATA SETS ARE USED AS INPUT TO STAND ALONE UTILITIES.

Explanation: A page in an IMAGE COPY data set has a zero header. Pages with zero headers are illegal if image copy data sets are used as input to stand-alone utilities.

Severity: 4 (warning)

System action: Processing continues.

User response: If the image copy data set name was invalid, look in SYSIBM.SYSCOPY for valid names. Resubmit the valid IMAGE COPY data set to the stand-alone utility. If the IMAGE COPY data set had a valid name, use the REPAIR utility to replace the zero pages with valid data pages.

DSN1964I NO VALUE HAS BEEN SPECIFIED FOR THE NUMBER OF PARTITIONS.

Explanation: No value was given for the number of partitions. This value should follow the 'NUMPARTS' parameter.

Severity: 8 (error)

System action: Processing will terminate.

User response: Look in SYSIBM.SYSTABLESPACE for the correct number of partitions for this page. Specify the correct number of partitions and resubmit the job. If the page is not partitioned, then either omit the NUMPARTS parameter or specify NUMPARTS as zero.

DSN1965I THE PARTITION NUMBER VALUE IS INVALID.

Explanation: The partition number value following the 'NUMPARTS' parameter is not between 1 and 64.

Severity: 8 (error)

System action: Processing will terminate.

User response: Look in SYSIBM.SYSTABLESPACE for the correct number of partitions for this page.

DSN1966I OUTPUT DATA SET odsname NOT FOUND

Explanation: DB2 cannot find 'odsname' because it has not yet been defined.

Severity: 8 (error)

System action: Processing is terminated.

User response: Use access method services to define 'odsname' and any other VSAM data sets that have not yet been defined.

DSN1967I THE OBID ARRAYS ARE FULL

Explanation: More than 1000 records of OBIDs were
specified in the SYSXLAT file.

Severity: 8 (error)

System action: Processing stops.

Problem determination: DSN1COPY cannot be used
for table spaces containing more than 1000 tables.

DSN1968I OBIDS SHOULD NOT BE DUPLICATED

Explanation: There should not be any repeated source OBIDs or target OBIDs within the SYSXLAT file.

Severity: 8 (error)

System action: Processing is terminated.

User response: Change the SYSXLAT file.

DSN1969I SYSXLAT FILE CAN NOT BE OPENED

Explanation: The QSAM OPEN for the SYSXLAT file failed because no DD statement was provided.

Severity: 8 (error)

System action: Processing is terminated.

User response: Notify system programmer.

System programmer response: Provide a SYSXLAT DD statement if OBID translate is to be done. Otherwise, remove the SYSXLAT parameter and rerun.

Problem determination: Follow standard QSAM diagnostic procedures.

DSN1970I DATA PAGE CONTAINING OBID xxxx, HAS NO MATCHING VALUE IN THE SYSXLAT FILE

Explanation: One of the OBID records of the SYSXLAT file contains a source OBID value that does not match the OBID value of the source system.

Severity: 8 (error)

System action: Processing is terminated.

User response: Provide that record of the SYSXLAT file which contains an unmatched OBID value with an OBID that matches the OBID value of the source system.

This error can also be caused when an OBIDXLAT is attempted on an incremental image copy of a compressed segmented table space. For more information on how to proceed, see the OBIDXLAT

option description under DSN1COPY in *DB2 Utility Guide and Reference*.

DSN1971I DBID xxxx NOT FOUND IN THE SYSXLAT FILE. REPLACE SOURCE DBID xxxx OF THE SYSXLAT FILE WITH IT.

Explanation: The first record of the SYSXLAT file has the incorrect source DBID value.

Severity: 8 (error)

System action: Processing is terminated.

User response: Provide the first record of the SYSXLAT file with a source DBID value which matches the DBID value of the source system.

DSN1972I PSID xxxx NOT FOUND IN THE SYSXLAT FILE REPLACE SOURCE DBID xxxx OF THE SYSXLAT FILE WITH IT

Explanation: The second record of the SYSXLAT file has the incorrect source PSID value.

Severity: 8 (error)

System action: Processing is terminated.

User response: Provide the second record of the SYSXLAT file with the correct PSID value.

DSN1973I SOURCE DBID IS MISSING

Explanation: The first record of the SYSXLAT file contains no DBID value.

Severity: 8 (error)

System action: Processing is terminated.

User response: Provide the first record of the SYSXLAT file with a pair of source and target DBID values which match the DBID values of the source and target systems.

DSN1974I TARGET DBID IS MISSING

Explanation: The first record of the SYSXLAT file contains no target DBID value.

This message is issued by the following CSECT: DSN1SYSX

Severity: 8 (error)

System action: Processing is terminated.

User response: Provide the first record of the SYSXLAT file with a target DBID value which matches the DBID value of the target system.

DSN1975I SOURCE PSID IS MISSING

Explanation: The second record of the SYSXLAT file contains no PSID value.

Severity: 8 (error)

System action: Processing is terminated.

User response: Provide the second record of the SYSXLAT file with a pair of source and target PSID values which match the PSID values of the source and target systems.

DSN1976I TARGET PSID IS MISSING

Explanation: The target PSID is missing from the second record of the SYSXLAT file.

Severity: 8 (error)

System action: Processing is terminated.

User response: Provide the second record of the SYSXLAT file with a target PSID value which matches the PSID values of the target system.

DSN1977I SOURCE OBID IS MISSING

Explanation: There is a record in the SYSXLAT file with no OBID value.

Severity: 8 (error)

System action: Processing is terminated.

User response: Correct the SYSXLAT file so that all the records following the first and second records will contain a pair of source and target OBID values.

DSN1978I TARGET OBID IS MISSING

Explanation: There is a record in the SYSXLAT file with no target OBID value specified.

Severity: 8 (error)

System action: Processing is terminated.

User response: Provide the SYSXLAT file with an OBID value that matches the OBID value of the target system.

DSN1979I *csect-name* **SOURCE & TARGET VALUES MUST BE WITHIN THE RANGE OF 1 TO 65535.**

Explanation: The SOURCE and TARGET values of the SYSXLAT file must be between 1 and 65535.

Severity: 8 (error)

System action: Processing is terminated.

User response: Provide the SYSXLAT file with values that are between 1 and 65535.

DSN1980I BROKEN PAGE BIT SET, WILL RESET AND RETRY BROKEN PAGE TEST

Explanation: This message indicates that a broken page was detected. If message DSN1990I is issued following this message, determine the cause of the broken page. If message DSN1990I is not issued, reset the broken page bit with the REPAIR utility to correct the problem. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Severity: 4 (warning)

System action: Processing continues.

Operator response: Notify the system programmer.

System programmer response: Use either the RECOVER or REPAIR utility to correct the broken page. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination should have been done when the dump associated with the setting of the broken page bit was requested.

DSN1981I INVALID PARAMETER FIELD: bad-parameter

Explanation: An invalid input parameter was detected on the EXEC statement. The parameter is either not a valid keyword or it has invalid subparameters associated with it.

Examples of invalid subparameters are:

- The NUMPARTS subparameter was other than '(nn)'. 'nn' is a decimal number.
- The NUMPARTS subparameter was missing either a left or right parenthesis.
- The PRINT subparameter was missing either a left or right parenthesis.
- The PRINT subparameter specified a digit in a page number which was not hexadecimal.
- The PRINT subparameter contained a starting page number that is larger than the ending page number.
- The PRINT subparameter contains a page number larger than six hexadecimal digits.

Severity: 8 (error)

System action: Processing terminates.

User response: Correct the indicated parameter on the EXEC DD statement, and resubmit the job.

Problem determination: Refer to DSN1COPY, DSN1PRNT, or DSN1COMP documentation for parameter requirements.

DSN1982I NO BLANKS ARE ALLOWED WITHIN THE PARAMETER LIST

Explanation: There is a space or spaces between one or more parameters in the utility list.

Severity: 8 (error)

System action: Processing is terminated.

User response: Check the parameter list and make sure that there are no spaces between any of the parameters.

DSN1983I *csect-name* **DICTIONARY IS INVALID, REASON= X'***nnnn***'**

Explanation: The dictionary found in the data set is invalid. A DB2 internal error occurred during the validation of the expansion dictionary.

Severity: 8 (error)

System action: Processing terminates.

User response: Verify that the correct data set is specified. You can delete the EXPAND option and resubmit the job.

System programmer response: Determine the failing environment. Refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: The REASON identifies the nature of the error. The REASON is used for diagnostic purposes by the IBM Support Center.

DSN1984I UNEXPECTED PAGE NUMBER, EXPECTING: expected-page-number

Explanation: This message indicates that an unexpected page number was encountered. This message may be received if the input data set is an image copy but the FULLCOPY or INCRCOPY parameters were not specified. The 'expected page number' that is displayed (in hexadecimal) is the page number that was expected, not the one that was encountered.

Severity: 4 (warning) or 8 (error)

System action: Processing is terminated.

User response: Notify the system programmer.

System programmer response: Use DSN1PRNT to dump the table space or index space. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSN1985I ZERO PAGES ENCOUNTERED. FIRST PAGE = first-zero-page-number, LAST PAGE = last-zero-page-number

Explanation: One or more pages zero pages were encountered with valid data pages after them. A zero page is a page of all zeros or a page with only the page-synchronization bit on in the first and last byte of the page. The 'first zero page number' and the 'last zero page number' displayed in hexadecimal are the pages where zeros were first encountered and last encountered. In most cases, this is not an error condition. This is a normal condition for segmented table spaces and also for index spaces created with a FREEPAGE parameter not equal to zero. In addition, it is also normal for nonsegmented table spaces to contain zero pages after certain update scenarios terminated abnormally and data rollback was required.

Severity: 4 (warning)

System action: Processing continues.

User response: Notify the system programmer.

System programmer response: If the zero page is the result of segmenting, no action is necessary. Similarly, if the zero page is a valid zero page that was introduced by DB2, no action is necessary as this page is free space and is used by DB2 when needed. If the zero page is a valid zero page and you want to remove it from the table space, run the REORG utility.

To help determine if the zero page was validly introduced by DB2, run the CHECK INDEX utility. This assumes that at least one index is defined on the affected table space.

If the zero page is determined to be invalid, you need to run the RECOVER utility.

DSN1986I NEITHER FULLCOPY NOR INCRCOPY PARAMETERS MAY BE SPECIFIED WITH THE SEGMENT PARAMETER.

Explanation: The parameters SEGMENT, FULLCOPY, and INCRCOPY describe the type of input that SYSUT1 references. SEGMENT is only relevant when a table space or index space is the input. FULLCOPY and INCRCOPY are only relevant when a image copy is the input. If the input data set is a segmented table space, then the FULLCOPY or INCRCOPY parameters should not be specified. If the input data set is a full or incremental image copy, then SEGMENT should not be specified.

Severity: 8 (error)

System action: Processing will terminate.

User response: Omit the incorrect parameter.

DSN1987I *csect-name* **INDEX PAGE CONTAINING OBID** *id* **HAS NO MATCHING VALUE IN THE SYSXLAT FILE.**

Explanation: This message is issued when the OBID on the page does not find a matching value in the input source OBID list. This message can also be issued when the input does not have enough OBID values.

Severity: 8 (error)

System action: Processing terminates with a dump of the page on which the OBID does not find a match in the source list.

User response: Resubmit the job with correct input OBID values.

DSN1988I *csect-name* **EXPAND KEYWORD IS SPECIFIED BUT DICTIONARY IS NOT FOUND**

Explanation: DSN1PRNT issues this message when the EXPAND option is specified. The dictionary was not found in the data set. An incorrect data set piece is specified or an incorrect data set number is specified for a multi-piece linear data set.

Severity: 8 (error)

System action: Processing terminates.

User response: Resubmit the job specifying the correct keyword or the correct data set name.

DSN1989I *csect-name* **IS PROCESSED WITH THE FOLLOWING OPTIONS:**

- *check/32K/fullcopy/incrcopy/segment/*
- *numparts=n/format/expand/swonly/*
- *print/value/obidxlat/reset/no check/4K/*
- *no imagecopy/nonsegment/numparts=0/*
- *no print/no value/no obidxlat/*
- *no format/no reset/no expand/no swonly*

Explanation: The stand-alone utility named *csect-name* is processing with the options that are either specified or supplied by default.

Severity: 0 (informational)

System action: Processing continues.

User response: Make sure that the parameters are correct. Not specifying the parameters correctly can give unpredictable results.

DSN1990I *csect-name* **BROKEN PAGE. PAGE NUMBER =** *broken-page-number***. REASON:** *comment***, SERVICE INFO: CODE =** *c1, d1* **=** *v1* **...** *dn* **=** *vn*

Explanation: During page checking, the page was detected as broken for the reason given in the message.

The page number is displayed in hexadecimal format.

Severity: 4 (warning)

System action: Processing continues.

Operator response: Notify the system programmer.

System programmer response: Use either the RECOVER or REPAIR utility to correct the broken page.

Problem determination: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Service information is used for diagnostic purposes by the IBM Support Center.

DSN1991I UNCLUSTERED DATA DETECTED. RID: 'xxxxxxxx'X TABLE: tablename INDEX KEY: index-key

Explanation: Data is not broken. Only the arrangement of data on one of the following DB2 Catalog table spaces: SYSDBASE, SYSPLAN, and SYSVIEWS has been affected. Normally, data on these pages is arranged to minimize locking contention. DSN1COPY found that the arrangement of data no longer minimizes locking contention for the objects named by the INDEX KEY for the given TABLE. Therefore, an increase in the number of deadlocks and timeouts may be seen on SYSDBASE, SYSPLAN, and SYSVIEWS.

System action: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: Contact your IBM Support Center for assistance. Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 2, 5, 64.

DSN1992I VSAM PUT ERROR, RPLERREG = *return-code*, **RPLERRCD =** *reason-code*

Explanation: VSAM PUT failed with the indicated error code.

Severity: 8 (error)

System action: Processing terminates.

User response: Notify system programmer.

System programmer response: Follow standard VSAM diagnostic procedures. See *z/OS DFSMS: Macro Instructions for Data Sets* for more information on the standard VSAM procedures, the RPL fields, and the VSAM control blocks.

DSN1993I DSN1COPY/DSN1PRNT/DSN1COMP TERMINATED, *page-count* **PAGES PROCESSED**

Explanation: This message indicates that a stand-alone utility has not successfully completed processing. The page count is displayed in decimal format.

Severity: 8 (error)

System action: Processing terminates.

User response: Examine the previous error message of the application program.

System programmer response: Correct the previous error and resubmit the job.

DSN1994I DSN1COPY/DSN1PRNT/DSN1COMP COMPLETED SUCCESSFULLY, *page-count* **PAGES PROCESSED**

Explanation: This message indicates that a stand-alone utility successfully completed processing. The page count is displayed in decimal format.

Severity: 0 (informational)

System action: Processing terminates.

DSN1995I VSAM GET ERROR, RPLERREG = *return-code*, **RPLERRCD =** *reason-code*

Explanation: VSAM GET failed with the indicated error code.

Severity: 8 (error)

System action: Processing terminates.

User response: Notify system programmer.

System programmer response: Follow standard VSAM diagnostic procedures. See *z/OS DFSMS: Macro Instructions for Data Sets* for more information on the standard VSAM procedures, the RPL fields, and the VSAM control blocks.

DSN1996I VSAM OPEN ERROR, ACBERRFLG = ACB *error-code*. **OPEN** *option*

Explanation: VSAM OPEN failed with the indicated 'ACB error-code'. The OPEN option was either 'INPUT' or 'OUTPUT'.

Severity: 8 (error) if error is greater than or equal to 128; otherwise, severity code is 4.

System action: Processing terminates if error code is greater than or equal to 128; otherwise, processing continues normally.

User response: Notify system programmer.

System programmer response: Follow standard VSAM diagnostic procedure. See *z/OS DFSMS: Macro Instructions for Data Sets* for more information on the

standard VSAM procedures and the VSAM control blocks.

DSN1997I **OUTPUT DSNAME =** *data-set-name*, *data-set-type*

Explanation: The indicated 'data set name' is used for output. 'data set type' is either VSAM or SEQ (for sequential).

Severity: 0 (informational)

DSN1998I **INPUT DSNAME =** *data-set-name*, *data-set-type*

Explanation: The indicated 'data-set-name' is used for input. 'data-set-type' is either VSAM or SEQ (indicating sequential).

Severity: 0 (informational)

DSN1999I **START OF DSN1COPY/DSN1PRNT/ DSN1COMP FOR JOB** *job-name step-name*

Explanation: This message indicates that a stand-alone utility started processing.

Severity: 0 (informational).

System action: The stand-alone utility starts processing.

Chapter 25. DSN2... messages

There are two versions of the CICS attachment facility:

- The version shipped with DB2 requires CICS Version 3 Release 3 or an earlier release. This attachment facility issues DSNCxxx messages.
- The version shipped with CICS requires a CICS release later than Version 3 Release 3. This attachment facility issues DSN2xxx messages.

DSN2001I CICS UR INDOUBT RESOLUTION IS INCOMPLETE FOR *ssid*

Explanation: CICS cannot resolve the disposition of the units of recovery that DB2 holds indoubt from the previous connection because CICS was cold started.

System action: The CICS attachment facility continues the startup process.

Operator response: Notify the CICS system programmer.

System programmer response: You must resolve the indoubt units of recovery manually by issuing the DB2 -DISPLAY THREAD and -RECOVER INDOUBT commands.

DSN2002I *csect-name* **DSN2CT***xx* **MODULE NOT FOUND**

Explanation: The resource control table DSN2CT with suffix *xx* could not be loaded.

System action: The CICS attachment facility initialization terminates.

User response: Reenter the start command with the correct suffix. Ensure that the resource control table DSN2CT is in the correct application program library, which is concatenated in the JOBLIB or STEPLIB statement of your CICS startup JCL.

If DSN2CT is not in your application program library, ensure that you are using the correct version of the DSNCRCT macro to assemble DSN2CT. For CICS versions later than Version 3 Release 3, the correct DSNCRCT macro is in the CICS macro library rather than in the DB2 macro library.

DSN2003I THE ATTACHMENT FACILITY IS ALREADY ACTIVE

Explanation: The CICS attachment facility is already active and enabled to CICS.

System action: The CICS attachment facility initialization is terminated.

Operator response: Only one CICS attachment facility is permitted.

DSN2004I THE PLAN NAME IS NOT FOUND IN THE RCT

Explanation: This message is the response to a DSNC DISPLAY or DSNC DISCONNECT command. The plan specified in the command was not found in the resource control table (RCT).

System action: The CICS attachment facility command is not processed.

Operator response: Reenter the command with the correct plan name.

DSN2005I THE TRANSACTION NAME IS NOT FOUND IN THE RCT

Explanation: This message is the response to a DSNC DISPLAY or DSNC MODIFY command. The transaction specified in the command was not found in the resource control table (RCT).

System action: The CICS attachment facility command is not processed.

Operator response: Reenter the command with the correct transaction name.

DSN2006I THE *dest* **DESTINATION ID IS INVALID**

Explanation: This message is the response to a CICS attachment facility command. The destination ID *dest* specified in the command was not found in the CICS destination control table (DCT) or the resource control table (RCT).

System action: The CICS attachment facility command is not processed.

Operator response: Reenter the command with the correct destination ID.

DSN2007I THE COMMAND VERB IS MISSING OR INVALID

Explanation: The CICS attachment facility does not recognize the ACTION keyword on the entered command.

System action: The command is not processed.

© Copyright IBM Corp. 1983, 2008

Operator response: Reenter the command with the correct syntax.

DSN2008I KEYWORD IS MISSING OR INVALID

Explanation: A CICS attachment facility command contains an unknown positional keyword or a keyword is missing.

System action: The command is not processed.

Operator response: Reenter the command with the correct syntax.

DSN2009I THE VALUE IN THE COMMAND IS INVALID

Explanation: The numeric value in the DSNC MODIFY command might be invalid. Otherwise, the error is caused by one of the following:

- The value in the command is greater than the value specified for the THRDM parameter in the resource control table (RCT).
- If the POOL RCT entry is being changed, the active thread count cannot be set to less than 3.

System action: The command is not processed.

Operator response: Reenter the command with a valid value.

System programmer response: If the number of threads required is more than the number allowed in the resource control table (RCT), specify a larger number of threads in the THRDM parameter and regenerate the table.

DSN2011I *tran* **TRANSACTION ABENDED BECAUSE THREADS ARE UNAVAILABLE**

Explanation: The transaction with CICS transaction ID *tran* was abnormally terminated because a thread connection to DB2 was not available for the transaction. This error message can occur when:

- TWAIT of the transaction entry in the resource control table is NO.
- TWAIT of the transaction entry is POOL, TWAIT in the pool entry is NO, and all threads in the POOL are active.

System action: The indicated transaction is abended. No other action is taken by the CICS attachment facility

Operator response: Notify the CICS system programmer.

System programmer response: Determine which of the following actions will correct the problem:

- If more threads are available with the current generation of the resource control table (RCT), tell the master terminal operator to make more threads active by using the DSNC MODIFY command.
- Regenerate the RCT and make more threads available to the transaction that failed. Stop and then restart the CICS attachment facility to make the new RCT active.

DSN2012I THE ATTACHMENT FACILITY STOP QUIESCE IS PROCEEDING

Explanation: A DSNC STOP command entered by the master terminal operator is being processed.

System action: The CICS attachment facility releases the terminal entering the command and proceeds with the quiesce process.

Operator response: If the quiesce is not completed within the required time period, you can enter a DB2 -STOP FORCE command.

DSN2013I DISPLAY REPORT FOLLOWS

Explanation: A CICS attachment facility display was requested and follows this message.

System action: Processing continues.

DSN2014I STATISTICS REPORT FOR *rct-name* **FOLLOWS**

Explanation: A CICS attachment facility statistics display was requested and follows this message. The name of the resource control table (RCT) that is currently in use is *rct-name*.

The following information is displayed for each entry in the RCT:

TRAN CICS transaction name.

PLAN DB2 plan name.

CALLS Total number of SQL statements in this connection.

AUTHS
Total number of sign-on invocations for transactions associated with this entry. A sign-on does not indicate whether a new thread is created or an existing thread is reused. If the thread is reused and TOKENE=NO, a sign-on occurs only if the authorization ID or transaction ID has changed.

W/P Number of times all available threads for this entry were busy and the transaction had to wait or the thread request was diverted to the POOL.

HIGH Maximum number of concurrent threads required by transactions associated with this entry at any time since the CICS attachment facility was started.

ABORTS
Total number of units of recovery that were rolled back.

COMMITS
One of the following two fields is incremented each time a DB2 transaction associated with this entry has an explicit or implicit (such as EOT) sync point.

1-PHASE
The total number of single phase commits for transactions associated with this entry.

2-PHASE
The total number of two phase commits for transactions associated with this entry.

For a more detailed explanation of the fields in the statistics report, refer to "Controlling CICS Connections" in Part 4 (Volume 1) of *DB2 Administration Guide*.

System action: Processing continues normally.

DSN2015E *csect* **THE ATTACHMENT FACILITY FOR** *name* **IS IN STANDBY**

Explanation: The CICS attachment facility has dropped into standby mode because DB2 has stopped and STANDBY=SQLCODE was specified in the RCT.

System action: The CICS attachment facility will issue negative SQLCODEs for all SQL requests.

Operator response: Restart the attachment after DB2 is restarted.

DSN2016I *csect-name* **THE CONNECTION OF CICS WITH** *name* **FAILED, '***reason-code***'**

Explanation: The CICS attachment facility is terminating abnormally because an error occurred while connecting with DB2. The error is indicated by *reason-code*, and the specific DB2 subsystem is indicated by *name*.

System action: The CICS attachment facility terminates with abend completion code X'04E'. If the initial identify fails because of an authorization error, DSN2042I is issued and no dump is requested.

Operator response: Notify the system programmer.

System programmer response: Look up *reason-code* to determine the specific failure that caused the termination.

DSN2017I ATTACHMENT OF A THREAD SUBTASK FAILED

Explanation: The CICS attachment facility continues processing but is unable to start as many thread subtasks as are allowed in a resource control table (RCT) entry.

System action: The CICS attachment facility attempted to terminate all subtasks that are currently unused, but none could be terminated. If other subtasks are not active to process the current transaction, CICS is requested to abend the transaction.

Operator response: Notify the system programmer.

System programmer response: The error might be caused by one of the following:

- There is a shortage of storage in the local storage queue area (LSQA).
- The maximum number of thread subtasks allowed was reached. The maximum is specified in the THRDMAX parameter of RCT. The DSNC DISPLAY STATISTICS command is useful in analyzing the current thread usage.

DSN2018I *name* **IS NOT ACTIVE**

Explanation: The DSNC STRT command cannot proceed because subsystem *name* is not active.

System action: The DSNC STRT command processing ends without starting the CICS attachment facility.

Operator response: Reenter the CICS attachment facility command after DB2 is started.

DSN2019I THE MODIFY COMMAND IS COMPLETE

Explanation: The DSNC MODIFY command completed successfully.

DSN2020I THE DISPLAY COMMAND IS COMPLETE

Explanation: The DSNC DISPLAY command completed successfully.

DSN2021I THE DISCONNECT COMMAND IS COMPLETE

Explanation: The DSNC DISCONNECT command completed successfully.

DSN2022I THE ATTACHMENT FACILITY STOP FORCE IS PROCEEDING

Explanation: A DSNC STOP FORCE command entered by the master terminal operator is being processed.

System action: The CICS attachment facility releases the terminal entering the command and proceeds with the STOP FORCE processing.

DSN2023I THE ATTACHMENT FACILITY HAS CONNECTED TO *name* USING *rct-name*

Explanation: The DSNC STRT command successfully completed. Subsystem *name* is now available to CICS. The name of the resource control table (RCT) that was specified for this start of the CICS attachment facility is *rct-name*.

System action: The DSNC STRT command completes processing. The CICS attachment facility is active.

DSN2024I *dest* CANNOT BE LOCATED IN THE DCT

Explanation: The destination shown in the message cannot be located in the CICS destination control table (DCT).

System action: The CICS attachment facility startup process continues. The destination entry in the resource control table (RCT) is set to '*' for potential modification by the CICS attachment facility.

Operator response: Notify the system programmer. Use the DSNC MODIFY command to set the error destination to the correct identification characters.

System programmer response: Correct either the RCT or the DCT to synchronize the error destinations.

DSN2025I THE ATTACHMENT FACILITY IS INACTIVE

Explanation: This is the final message issued by the CICS attachment facility to inform the resource control table error destinations that the CICS attachment facility completed the shutdown process.

System action: The CICS attachment facility is terminated.

DSN2026I DSN2STOP FAILED TO DISABLE THE *name* RMI EXIT

Explanation: The CICS attachment facility failed to disable the named resource manager interface (RMI) exit. Either the exit was in use at the time DSN2STOP tried to disable it, or an abend occurred in the RMI exit code while the 'in use' flag was on. The CICS system must be stopped and restarted before the CICS attachment facility can be restarted.

System action: The CICS attachment facility continues termination processing after dumping the CICS trace, the resource control table (RCT), and the connection control table (CCT).

Operator response: Notify the CICS attachment facility system programmer. CICS must be stopped and restarted before restarting the CICS attachment facility.

System programmer response: This could be a problem in the CICS 'task-related user exit'. Analyze the trace and any dumps associated with this execution to determine the source of the error.

DSN2027I SHUTDOWN IS IN PROGRESS. *transaction* COMMANDS MAY NOT BE ENTERED

Explanation: The CICS attachment facility does not accept commands directed to DB2 during or after the termination of the CICS attachment facility.

System action: The command is rejected.

User response: Notify the CICS attachment facility system programmer.

DSN2028I USER IS NOT AUTHORIZED FOR *transaction* COMMANDS

Explanation: DB2 rejected the command request during the sign-on of the user. Therefore, the user is assumed to be unauthorized for the requested function in the command.

System action: The command is rejected.

Operator response: Notify the CICS attachment facility system programmer.

System programmer response: The authorization ID used is described in the generation procedures of the resource control table (RCT). If this user should be allowed access to the command, use the GRANT statement to grant authorization.

DSN2029I *transaction* COMMAND FAILED, REASON CODE *rc*

Explanation: The DB2 command, identified in the message as *transaction*, failed to return a response message indicating the action taken by the command. DB2 returned reason code *rc*.

System action: The results of the command are unknown.

Operator response: Notify the CICS attachment facility system programmer.

System programmer response: Check the master console for any messages indicating why the command failed. Refer to the explanation of the reason code returned by DB2 in *DB2 Codes*.

DSN2030E AN ERROR HAS OCCURRED WRITING TO THE TRANSIENT DATA OUTPUT DESTINATION *destination*

Explanation: The CICS attachment facility encountered an error while writing to the indicated destination.

System action: The CICS attachment facility continues operation.

Operator response: Notify the system programmer. If the destination is one of the resource control table error destinations, use the CICS MODIFY command to change the destination code to another destination, if one is available.

System programmer response: Follow CICS procedures to correct the error. Refer to *CICS Transaction Server for z/OS Problem Determination Guide* for additional information.

DSN2031I THE COMMAND IS INVALID. THE ATTACHMENT FACILITY IS INACTIVE

Explanation: STRT is the only valid command when the CICS attachment facility is inactive. No other commands to the CICS attachment facility are accepted.

System action: The command is rejected.

DSN2032I *transaction* **COMMAND ALTERNATE DESTINATION OUTPUT COMPLETE**

Explanation: The DB2 command identified in the message as *transaction* is complete. The resulting output should be available at the requested device.

System action: Processing continues normally.

DSN2033I TERMINAL *termid* **IS NOT SUPPORTED BY BMS OR IS INVALID**

Explanation: This message is issued in response to a CICS attachment facility or DB2 command that requested an alternate destination for the response. CICS basic mapping support (BMS) encountered an error while routing to the requested terminal named *termid*.

System action: The output from the command might be suppressed.

Operator response: Ensure that the terminal ID was correctly entered. Otherwise, notify the system programmer.

System programmer response: Verify that the terminal ID is valid. This message might occur if the destination device is not supported by BMS or if it is not defined to CICS. If *termid* is CSMT, this message might also follow message DSN2030E. The attachment facility unconditionally routes output to terminal CSMT following a transient data destination error.

DSN2034I INDOUBT RESOLUTION FOR *ur-id* **IS INCORRECT**

Explanation: CICS indicates recovery is required for the *ur-id*, but the unit of recovery is not indoubt to DB2.

System action: The CICS attachment facility continues the startup process.

Operator response: Notify the CICS system programmer.

System programmer response: CICS tried to recover a unit of recovery using an identifier that was not in the resolution request list from DB2. Use the DB2 -DISPLAY THREAD and -RECOVER INDOUBT commands to ensure that all indoubt units of recovery are resolved.

You must identify the reason for the inconsistency. A DB2 cold start, for example, might cause the problem.

DSN2035I INDOUBT RESOLUTION FOR *ur-id* **IS INCOMPLETE**

Explanation: CICS indicates recovery should not be required for the *ur-id*, but DB2 is indoubt.

System action: The CICS attachment facility continues the startup process.

Operator response: Notify the CICS system programmer.

System programmer response: CICS cannot recover a unit of recovery using an identifier that was in the resolution request list from DB2. Use the DB2 -DISPLAY THREAD and -RECOVER INDOUBT commands to ensure that all indoubt units of recovery are resolved.

You must identify the cause of the error. A CICS cold start could cause this problem.

DSN2036I INDOUBT RESOLUTION FOR *ur-id* **IS INCONSISTENT**

Explanation: CICS indicates recovery resolution for the *ur-id* should be ABORT (backout), but DB2 indicates the resolution should be COMMIT.

This error can be caused by a damaged CICS system journal file.

System action: DB2 committed the unit of recovery. The CICS attachment facility continues the startup process.

Operator response: Notify the CICS system programmer.

System programmer response: CICS and/or DB2 restart processes are not in synchronization. You must identify the cause of the error. Take appropriate action to return all resources to a consistent state.

DSN2037A *subsystem* **IS NOT ACTIVE** *transaction* **IS WAITING**

Explanation: The CICS attachment facility start transaction identified in the message is waiting for notification from the indicated DB2 subsystem that it is active. The start of the CICS attachment facility is complete after DB2 has started.

System action: The CICS attachment facility waits to complete the start process.

Operator response: Start the DB2 subsystem.

DSN2038I THE COMMAND IS INVALID WHILE WAITING FOR *subsystem name*

Explanation: The command entered cannot be processed while the CICS attachment facility is waiting for the DB2 subsystem to start. The name of the DB2 subsystem that is not yet operational is *subsystem name*.

System action: The command is rejected.

DSN2039I THE ERROR DESTINATIONS ARE: *d1 d2 d3*

Explanation: This message is the response to the DSNC MODIFY DESTINATION command that lists the currently active message destinations known to the CICS attachment facility. Null entries that show as '****' can be modified so they identify actual destinations.

DSN2040I MODULE *name* **COULD NOT BE FOUND**

Explanation: During the CICS attachment facility initialization, an attempt was made to locate and load the named module, but it was not found in any of the libraries accessible to MVS through the MVS LOAD macro.

System action: The CICS attachment facility initialization is terminated.

Operator response: Notify the CICS attachment facility system programmer.

System programmer response: Check the CICS JOBLIB/STEPLIB libraries and ensure that the CICS SDFHAUTH and the DB2 SDSNLOAD libraries are defined there.

DSN2041I NO ACTIVE THREADS

Explanation: A DSNC DISPLAY TRANSACTION or DSNC DISPLAY PLAN command was entered, but there were no active threads created. The CICS attachment facility might have identified and signed on the attached subtasks, but a create thread was not issued for any of the subtasks.

DSN2042I CONNECTION NOT AUTHORIZED TO *name*

Explanation: The attempt to connect to the named DB2 subsystem failed because the user was not authorized to access DB2. Authorization was denied by either RACF or a user-written connection exit.

System action: The CICS attachment facility terminates.

Operator response: Notify the system programmer.

System programmer response: The user specified on the CICS job was not authorized to connect to the named DB2 subsystem. Refer to Part 3 (Volume 1) of *DB2 Administration Guide* for information on how to authorize a user to access DB2.

DSN2043I DSN2STOP ENCOUNTERED ERROR, RECOVERY ROUTINE started/completed

Explanation: A CICS abend occurred while disconnecting the CICS attachment facility from DB2 and deleting it from storage. This message might be issued twice. It is issued at the start of the abend recovery routine, which is indicated by 'started' in the message text. If the recovery routine is successful, the message is issued with 'completed' in the message text.

System action: If the recovery routine is not successful, program DSN2STOP terminates with a CICS transaction dump. If the recovery routine is successful, the CICS attachment facility is terminated normally.

Operator response: Notify the system programmer.

System programmer response: Analyze any abend dumps associated with this message to determine the cause of the failure.

DSN2044I DSNCRCT ENTRY HAS BEEN CORRUPTED OR OVERLAID

Explanation: The CICS attachment facility detected that a resource control table (RCT) entry was corrupted or overlaid since the CICS attachment facility was last started.

System action: The CICS task is abended with reason code 00C30011. No other action is taken by the CICS attachment facility.

Operator response: Notify the system programmer.

System programmer response: Analyze the abend dumps associated with this message to determine which CICS transaction and RCT entry was corrupted. Restart the CICS attachment facility to obtain a valid copy of the RCT.

DSN2045I RESOURCE MANAGER *rmi-name* **IS UNKNOWN TO CICS ATTACH FACILITY**

Explanation: The CICS attachment facility received a request from a resource manager interface (RMI) named *rmi-name*. This RMI is not known by the CICS attachment facility.

System action: The CICS task is abended with a CICS transaction dump.

Operator response: Notify the system programmer.

System programmer response: Analyze the CICS transaction dump to determine the source of the unknown resource manager interface (RMI) request.

Problem determination: CICS dump information, such as the trace table and program storage, can aid in determining which module issued the request. The CICS EXEC interface block resource name field, EIBRSRCE, contains the name of the RMI.

If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSN2046I RCT RELEASE LEVEL *nnnn* IS INCONSISTENT WITH CICS ATTACH RELEASE LEVEL *mmmm*

Explanation: The resource control table (RCT) release level *nnnn* does not match the CICS attachment facility release level *mmmm*.

This message is issued by the following CSECT: DSN2STRT

Operator response: Notify the system programmer.

System programmer response: Reassemble the RCT using the correct DB2 release level *mmmm* DSNCRCT macro.

DSN2047I TWA SIZE IS TOO SMALL FOR TRANSID *tran*

Explanation: The CICS TWASIZE defined for transaction *tran* is too small to support the CICS attachment facility command processor program (DSN2COM1).

System action: The CICS attachment facility abends this CICS task with a CICS transaction abend.

Operator response: Notify the system programmer.

System programmer response: Redefine the command processor transaction to specify the correct TWASIZE.

DSN2048I UNABLE TO INTERPRET SQL CALL WHILE FORMATTING AN EDF DISPLAY TRAN *tran* TASK *task*

Explanation: The CICS attachment facility was unable to decipher storage associated with an SQL statement for CICS transaction *tran* and task id *task*. Either the SQL communications area (SQLCA) or the RDS input parameter list (RDI) storage could not be interpreted by the CICS attachment facility.

System action: A CICS transaction dump is taken of the storage in question.

Programmer response: Ensure that CICS storage associated with this task was not altered or overlaid. Analyzing the storage in the CICS transaction dump might provide clues about the source of the problem.

Problem determination: Analyze the CICS transaction dump of the storage in question. In this situation the information supplied by the Execution Diagnostic Facility (EDF) of CICS for SQL statements consists of:

- The EDF status: ABOUT TO EXECUTE or COMMAND EXECUTION COMPLETE
- The processing status: CALL TO RESOURCE MANAGER DSNCSQL
- The ARG values associated with this call to the CICS attachment facility resource manager

No other information is provided about the SQL statement.

DSN2049E *csect-name* RECOVER INDOUBT FOR urid FAILED WITH *reason-code*

Explanation: A resolve indoubt request passed to DB2 from CICS failed.

System action: The attachment leaves the unit of recovery indicated by 'urid' indoubt in CICS and DB2. The attachment requests a system dump of the CICS region.

System programmer response: Use the reason code to determine why the RECOVER INDOUBT request failed. If necessary, manually resolve the unit of recovery using the RECOVER INDOUBT command.

Problem determination: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Collect the following diagnostic items listed in Appendix A, “Problem determination,” on page 659: 82,

---

DSN2052I INSUFFICIENT STORAGE AVAILABLE TO START ATTACHMENT FACILITY

Explanation: During startup the CICS attachment facility was unable to allocate the required amount of storage. This message might be issued if an error occurs while the attachment facility is loading the data only load module, DSN2CTxx.

System action: The CICS attachment facility is not initialized.

System programmer response: Allow a larger amount of dynamic storage on your CICS system. Alternatively, you can delete unused entries or lower THRDA and THRDM values in your RCT, which reduces the size of the storage that the attachment facility requires.

For a load error, check the MVS console log for any preceding messages to determine why the load failed. If there is not enough free REGION space to load the RCT, see the appropriate CICS documentation for information on tuning your CICS system.

DSN2053I INSUFFICIENT TWASIZE FOR DSN2COM1. REQUIREMENT IS *twasize*

Explanation: The CICS attachment facility requires a TWASIZE of *twasize* on all transactions that execute DSN2COM1. Either the TWASIZE that was specified is less than the requirement, or no TWASIZE was specified at all.

System action: The attachment facility terminates.

System programmer response: Ensure that CSD group DFHDB2 was installed in the CSD. Any transaction that executes program DSN2COM1 should specify TWASIZE(*twasize*) on its transaction entry.

DSN2054I DYNAMIC PLAN PROGRAM LINK FAILED FOR TRANSACTION '*transaction***'. EIBRESP WAS '***eibresp***'.**

Explanation: The attachment facility was unable to link to the dynamic plan exit program specified in DSNCRCT. The reason code provided by CICS in the EIBRESP field of the EIB is given in this message.

System action: The transaction is abended with a DSNC abend.

User response: Look up the *eibresp* value in *CICS Transaction Server for z/OS Application Programming Reference*. This value was returned from CICS after the attachment facility attempted to link to the dynamic plan exit. Take appropriate action based on the *eibresp*. For example, if *eibresp* was PGMIDERR, check your RCT entry for transaction *transaction* to ensure that the dynamic plan exit program name (PLNPGME or PLNPGMI) is the name of a valid program known to CICS.

DSN2055I SINGLE PHASE COMMIT FAILED FOR TRANSACTION '*transaction***'. EIBTASKN WAS '***task number***'.**

Explanation: CICS requested a single phase commit from DB2, but DB2 was unable to comply.

System action: The transaction is abended with a DSNC abend.

User response: Contact the IBM Support Center for assistance.

DSN2056I *csect-name* **CICS RELEASE** *release* **IS DOWN LEVEL FOR THIS ATTACHMENT FACILITY**

Explanation: The attachment facility that you are using requires a CICS release later than Version 3 Release 3. You are using CICS Version 3 Release 3 or an earlier release. For this release of CICS, you must use the attachment facility shipped with DB2.

System action: The attachment facility does not initialize.

User response: Ensure that the CSD definition of the DSNC transaction specifies program DSNCCOM1. If you are initializing the attachment facility using CICS program list table (PLT) processing, ensure that your CICS PLTPI table uses program DSNCCOM0 to initialize the attach.

DSN2059I *csect-name* **CICS ATTACHMENT FACILITY MISMATCH WITH RESOURCE CONTROL TABLE** *rct-name*

Explanation: The wrong macro library was used to assemble DSN2CT, the resource control table.

There are two versions of the CICS attachment facility. The version shipped with DB2 requires CICS Version 3 Release 3 or an earlier release; the one shipped with CICS requires a CICS release later than Version 3 Release 3. This error message indicates CICS is using the attachment facility shipped with CICS, but the RCT was assembled using the DB2 macro libraries.

System action: The CICS attachment facility initialization terminates.

User response: Reassemble DSN2CT using the DSNCRCT macro shipped in the CICS macro libraries.

DSN2060I *csect-name* **INITPARM IS IGNORED BECAUSE FORMAT IS INVALID.**

Explanation: The DSN2STRT INITPARM specified on the CICS initialization job is incorrect. The format of the INITPARM should be:

```
INITPARM=(DSN2STRT='xx,yyyy')
```

where xx is the suffix of the RCT and yyyy is the DB2 subsystem id. The RCT suffix cannot be greater than 2 characters, and the subsystem ID cannot be greater than 4 characters.

To specify an RCT suffix only, use the format:

```
INITPARM=(DSN2STRT='xx')
```

To specify a subsystem ID override only, use the format:

```
INITPARM=(DSN2STRT=',yyyy')
```

System action: The attachment facility attempts to initialize with the subsystem ID specified in the DSNC STRT command. If no subsystem ID is provided there, it attempts to initialize with the subsystem ID specified in the resource control table.

User response: If you want to override the subsystem ID, restart the attachment facility using the DSNC STRT command with the correct SSID. Alternatively, you can correct the DSN2STRT INITPARM and reinitialize CICS.

DSN2061I *csect-name* **INITPARM IS INVALID. ATTACHMENT FACILITY NOT STARTED.**

Explanation: No resource control table suffix was specified on the DSNC STRT command, and the format of the DSN2STRT INITPARM on the CICS initialization job is incorrect. The format of the INITPARM should be:

```
INITPARM=(DSN2STRT='xx,yyyy')
```

where xx is the suffix of the RCT and yyyy is the DB2 subsystem id. The RCT suffix cannot be greater than 2 characters, and the subsystem ID cannot be greater than 4 characters.

To specify an RCT suffix only, use the format:

```
INITPARM=(DSN2STRT='xx')
```

To specify a subsystem ID override only, use the format:

```
INITPARM=(DSN2STRT=',yyyy')
```

System action: The attachment facility does not initialize.

User response: To start the attachment facility, use the DSNC STRT command and specify a resource control table suffix on the command. Alternatively, you can correct the DSN2STRT INITPARM and reinitialize CICS.

DSN2063I *csect-name* **NOT AUTHORIZED TO START TRANSACTION** *transaction***. ATTACHMENT FACILITY NOT STARTED**

Explanation: The user was not authorized to start transaction *transaction*.

The CICS-DB2 attachment facility starts *transaction* during initialization. Before it starts *transaction*, it checks external security (such as RACF) using CICS's QUERY SECURITY command. When CICS returns the value NOTREADABLE for *transaction*, the attachment facility produces this message.

System action: The attachment facility does not initialize.

System programmer response: Identify the authorization ID of the user who started the attachment facility. Use your external security product to authorize that user to start *transaction*.

DSN2064I *csect-name transaction* **ABENDED. ATTACHMENT FACILITY IS TERMINATING.**

Explanation: The CICS-DB2 transaction *transaction* abended.

System action: The CICS-DB2 attachment facility terminates.

System programmer response: Identify the cause of the abend, correct it if necessary, and restart the attachment facility.

DSN2065I *csect-name* **INVALID LENGTH. DATA NOT DISPLAYABLE.**

Explanation: The module running under the CICS Execution Diagnostic Facility (EDF) attempted to display an input or output variable that had an incorrect length indicator.

System action: EDF processing continues, but the value of the variable is not displayed.

DSN2070I **csect ROLLBACK FAILED FOR TRANSACTION transaction EIBRESP2 = X'eibresp2'**

Explanation: The attachment facility was unable to issue a CICS ROLLBACK. The reason code provided by CICS in the EIBRESP2 field of the EIB is shown in *eibresp2*.

System action: The transaction is abended with a DSNC abend.

System programmer response: Look up the *eibresp2* value in CICS Transaction Server for z/OS Application Programming Reference. This value was returned from CICS after the attachment facility attempted a ROLLBACK requested by the ROLBE=YES option.

DSN2700I **DSN2SAMG** *message-id* **HAS BEEN ISSUED**

Explanation: This message is issued only if the requested message was not found in the message directory (DSN2MSGT).

System action: The system proceeds as though the requested message was issued.

Operator response: Notify the system programmer.

System programmer response: Refer to this manual for the message text.

Problem determination: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSN2901I **UNRECOVERABLE IOERROR IN PROGRAM DSN2COM1. THE ATTACHMENT FACILITY WILL SHUTDOWN.**

Explanation: An attempt to retrieve data stored for program DSN2COM1 failed when CICS returned an exceptional condition of IOERR (EIBRESP = X'11').

System action: The DB2-CICS attachment facility is shutdown.

Operator response: Notify the system programmer.

System programmer response: An IOERR occurs if an input or output error occurs during a RETRIEVE operation. Examine the CICS auxiliary trace records and the CICS system log to determine the cause of the IOERR.

Chapter 26. DSN3... messages

DSN3001I *csect-name* **ABNORMAL DISCONNECT FROM SUBSYSTEM INTERFACE**

Explanation: This message indicates that 'csect-name' detected that an online routine was still supporting SSI calls (IEFSSREQ) even though DB2 was nearly done terminating or no longer executing. This occurs with 'csect-name' DSN3RS0Xwhen the service address space has reached end-of-memory and neither normal termination nor online error recovery routines have successfully completed termination of the DB2 subsystem. This occurs with 'csect-name' DSN3SSTM when this condition is discovered during online termination.

System action: The connection is terminated. All IEFSSREQ requests are handled by the DB2 ERLY code until DB2 is restarted.

Problem determination: An SVC dump is requested. This SVC dump and associated SYS1.LOGREC entries provide information to determine which resource managers had not removed their support of particular SSI function codes.

For dumps requested by DSN3RS0X,all pertinent control blocks and stack storage are dumped. The primary control block of interest is the SSVT for DB2.

For DSN3SSTM, the DB2 'standard dump' facility is used to dump the information necessary for problem diagnosis. The primary control blocks of interest are the 'active' CONN blocks.

DSN3002I **RECOVERY INCOMPLETE FOR connection-name**

Explanation: At least one DB2 unit of recovery (UR), related to an identified subsystem ('connection-name'), is still 'indoubt' after restart synchronization has taken place. ('Indoubt' URs are those for which commit has been voted by DB2 but which have not been acknowledged by 'connection-name'.)

This message may appear if the 'connection-name' subsystem has begun to do new work before having resolved all indoubt URs. The 'connection-name' subsystem may be in the process of resolving the indoubt URs.

System action: Resources held (locked) by these 'indoubt' URs are unavailable to any other work units until their status is resolved.

System programmer response: The system programmer and/or system administrator must determine the correct recovery action to resolve the 'indoubt' situations. This involves either 'ensure-commit' or 'backout' decisions for all 'indoubt' URs.

The -DISPLAY THREAD command should be used to see the URs still indoubt. It may show that all indoubt URs have already been resolved. The -RECOVER INDOUBT command should be used, either from an MVS console or through an attachment facility, to resolve the indoubt URs and to release the resources they hold.

Problem determination: This error is probably caused by a cold start after an abnormal termination of either subsystem or by offline alterations of the logs of either subsystem.

DSN3004I **DESCRIPTOR GET FAILURE RC=X xxxxxxxx, REASON=X xxxxxxxx**

Explanation: The parameter list required to create the DB2 agents under which IEFSSREQ calls are processed could not be obtained from the subsystem support subcomponent directory of internal parameters (DSN3DIR1). The RC indicates the return code received from the system parameter manager and REASON indicates the reason code (from register 0).

The return code is always 4 when this message is presented, indicating that the IEFSSREQ descriptor was not found in the subsystem support subcomponent directory. This is a system error and results in termination of the DB2 subsystem. The reason code is always 0.

System action: Subsystem initialization is discontinued. Termination occurs.

System programmer response: Repair the master directory (DSNZPARM) and/or the DB2 program libraries, and restart the DB2 subsystem.

Problem determination: If the problem persists, use any dumps and the console log to diagnose the problem. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSN3006I **xxxx SSI FUNCTION WAS ALREADY ACTIVE WHEN ACTIVATE WAS ATTEMPTED**

Explanation: An initialization sequence error has occurred. The subsystem support for SSI services resource manager (SSSS) has discovered during its initialization that another resource manager ('xxxx') has already activated the EOM or FEOT SSI call by calling DSN3AC0X. If this occurs, the error is a serious one.

System action: Subsystem initialization is

© Copyright IBM Corp. 1983, 2008

discontinued. DB2 termination occurs.

System programmer response: Ensure that all maintenance has been applied to the DB2 program libraries, and then restart the DB2 subsystem.

Problem determination: If the problem persists, use any dumps and the console log to diagnose the problem. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSN3007I *xxxx* **SSI FUNCTION WAS ALREADY INACTIVE WHEN DEACTIVATE WAS ATTEMPTED**

Explanation: A termination sequence error has occurred. The subsystem support for SSI services resource manager (SSSS) has discovered during its termination that another resource manager ('xxxx') has already deactivated the EOM or FEOT SSI call by calling DSN3AC0X. If this occurs, the error is a serious one.

System action: Termination continues.

System programmer response: Ensure that all maintenance has been applied to the DB2 program libraries.

Problem determination: If the problem persists, use any dumps and the console log to diagnose the problem. If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

DSN3008I *csect-name* **ABNORMAL DISCONNECT FOR PROGRAM REQUEST HANDLER(S)**

Explanation: This message indicates that 'csect-name' has discovered that one or more resource managers are still supporting application program calls through their program request handler, even though DB2 was almost done terminating or was no longer executing. This occurs when the system services address space has gone to end of memory and neither normal termination nor online error recovery routines have successfully completed termination of the DB2 subsystem.

System action: The program request handler (PRH) connection is terminated. All application program support requests are rejected with an indication that DB2 is not up.

System programmer response: See the Problem Determination section of this message.

Problem determination: An SVC dump is requested. This SVC dump and associated SYS1.LOGREC entries provide information to determine which resource managers were still supporting application program support calls. All pertinent control blocks and stack storage are dumped. The primary control block of interest is the program request handler block (PRHB).

DSN3010I *csect-name* **PARTICIPANT RRS IS COLD STARTING AND HAS LOST ITS LOG. OTHER RECOVERABLE RESOURCE MANAGERS MIGHT HAVE INDOUBT UNITS OF RECOVERY.**

Explanation: DB2 has coordinator responsibility for indoubt units of recovery and RRS has a participant role in the unit of recovery. RRS has cold started and has lost knowledge of indoubt units of recovery. There might be damage at RRS Recoverable Resource Managers that also participated in the unit of recovery.

System action: Processing continues.

System programmer response: Contact the database administrator.

Database Administrator Action: DB2 is the coordinator for one or more indoubt units of recovery at other RRS recoverable resource managers. RRS has lost all knowledge of these units of recovery. The participant RRS Recoverable Resource Managers might still have indoubt units of recovery. DB2 has no knowledge of other RRS Recoverable Resource Manager except for the RRS unit of recover ID (URID) that identifies such a unit of recovery. Subsequent DSN3017I, DSN3018I or DSN3019I messages describe the status of each unit of recovery and contain the RRS URID for each.

DSN3011I *csect-name* **COORDINATOR RRS IS COLD STARTING AND HAS LOST ITS LOG. INDOUBT DB2 THREADS NEED MANUAL RESOLUTION.**

Explanation: DB2 has participant responsibility for indoubt threads. RRS, the commit coordinator, has informed DB2 that it lost all knowledge of DB2 indoubt threads. The indoubt threads at this DB2 must be manually resolved with the RECOVER INDOUBT command.

System action: Processing continues.

System programmer response: Contact the database administrator.

Database Administrator Action: DB2 is a participant with one or more indoubt threads. The coordinator is RRS. RRS has lost knowledge of the indoubt DB2 threads. A list of indoubt threads where RRS is the coordinator can be displayed using the DISPLAY THREAD command for indoubt type threads by specifying RRSAF as the connection name. A DSNV408I message with coordinator = RRSAF identifies a thread of this type. The DSNV449I message in the DISPLAY THREAD report that follows contains the RRS unit of recovery ID (URID) that identifies the RRS unit of recovery in the NID field.

The heuristic decision (that is, to commit or abort the logical unit of work) should be coordinated with any other participant RRS Recoverable Resource Managers. The existence of other participants might not be easy to

determine. The information might be available in the MVS/RRS recovery log even though information has been lost.

At this DB2, all indoubt threads coordinated by the RRS must be resolved with the RECOVER INDOUBT command. Locked data remains unavailable until resolution. Threads that were already resolved with this command are discarded. Threads not yet resolved are discarded after resolution with the command.

The commit or abort decision provided using the RECOVER INDOUBT command for a logical unit of work are propagated to all downstream participants, if any.

DSN3012I *csect-name* **DB2 WAS RESTARTED ON THE WRONG SYSTEM AND CANNOT CONNECT TO RRS. THERE ARE UNRESOLVED UNITS OF RECOVERY WHERE DB2 IS THE COORDINATOR.**

Explanation: DB2 is the commit coordinator for one or more indoubt units of recovery. DB2 is unable to connect to RRS to resolve these indoubt units of recovery.

System action: Processing continues.

Operator response: Use the RRS IPCS interface to determine the correct system and restart DB2 on that system.

DSN3013I *csect-name* **DB2 WAS RESTARTED ON THE WRONG SYSTEM AND CANNOT CONNECT TO RRS. THERE ARE UNRESOLVED UNITS OF RECOVERY WHERE DB2 IS A PARTICIPANT.**

Explanation: DB2 has one or more indoubt threads and is unable to connect to RRS to resolve these indoubt units of recovery.

System action: Processing continues.

Operator response: Use the RRS IPCS interface to determine the correct system and restart DB2 on that system.

DSN3014I *csect-name* **DB2 HAS NO KNOWLEDGE OF RRS URID=***rrs_urid***. THE RRS UR IS INDOUBT. THE DB2 URID RECORDED BY RRS=***db2_urid***.**

Explanation: DB2 is restarting with RRS where DB2 is a participant and RRS is the coordinator. RRS has a unit of recovery that DB2 should be a participant in, but DB2 has no knowledge of the RRS unit of recovery. The RRS unit of recovery ID is 'rrs_urid'. RRS has recorded the DB2 unit of recovery ID as 'db2_urid'.

System action: Restart with RRS continues.

Operator response: Notify the database administrator.

Database Administrator Action: This message might indicate a DB2 or RRS problem, or it might be produced normally because one of the following prior actions:

- A DB2 conditional restart was performed that resulted in the loss of part or all of the DB2 log. This conditional restart might have happened at any time in the past.
- The DB2 RECOVER INDOUBT command was used to resolve the DB2 unit of recovery with URID 'db2_urid'.

If either of these occurred, the message can be ignored. If neither occurred, there might be a DB2 or an RRS problem. If the 'db2_urid' appears to be a valid log RBA, use the DSN1LOGP utility with the SUMMARY option and URID options using the 'db2-urid' value. If DSN1LOGP finds the 'db2-urid', the disposition will indicate whether the DB2 UR was committed or aborted (rolled back). If possible, use the RRS ISPF interface to commit or abort the RRS URID to avoid heuristic damage.

Problem determination: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 6, 33. 72.

DSN3015I *csect-name* **DB2 HAS NO KNOWLEDGE OF RRS URID=***rrs_urid***. THE RRS UR IS INDOUBT. THE DB2 URID RECORDED BY RRS=***db2_urid***.**

Explanation: DB2 is restarting with RRS where DB2 is the coordinator and RRS is the participant. RRS has a unit of recovery that DB2 should be the coordinator of, but DB2 has no knowledge of the RRS unit of recovery. The RRS unit of recovery ID is 'rrs_urid'. RRS has recorded the DB2 unit of recovery ID as 'db2_urid'.

System action: Restart with RRS continues.

Operator response: Notify the database administrator.

Database Administrator Action: This message might indicate a DB2 or RRS problem, or it might be produced normally because a DB2 conditional restart was performed that resulted in the loss of part or all of the DB2 log. This conditional restart might have happened at any time in the past.

If the 'db2_urid' appears to be a valid log RBA, use the DSN1LOGP utility with the SUMMARY option and URID options using the 'db2-urid' value. If DSN1LOGP finds the 'db2-urid', the disposition will indicate whether the DB2 UR was committed or aborted (rolled back). Use the RRS ISPF interface to commit or abort the RRS URID to avoid heuristic damage.

Problem determination: If you suspect an error in

DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 6, 33. 72.

DSN3016I *csect-name* **RRS HAS LOST DATA FROM ITS LOG.**

Explanation: DB2 is restarting with RRS and RRS has lost some portion of its log.

System action: Restart with RRS continues.

Operator response: Notify the database administrator.

Database Administrator Action: DB2 might not be able to resolve indoubt Units of Recovery successfully with RRS because of the loss of RRS log data. This message might be followed by one or more of the following messages: DSN3017I, DSN3018I, DSN3019I, DSN3020I, DSN3021I, DSN3022I and DSN3023I.

DSN3017I *csect-name* **RRS HAS NO KNOWLEDGE OF DB2 URID=***db2_urid***. THE DB2 UR IS INDOUBT. THE RRS URID RECORDED BY DB2=***rrs_urid***.**

Explanation: DB2 is restarting with RRS where DB2 is the coordinator and RRS is a participant. DB2 has a unit of recovery that RRS should be a participant in, but RRS has no knowledge of the DB2 unit of recovery. The DB2 unit of recovery ID is 'db2_urid'. DB2 has recorded the RRS unit of recovery ID as 'rrs_urid'.

System action: Restart with RRS continues.

Operator response: Notify the database administrator.

Database Administrator Action: This message might indicate a DB2 or RRS problem, or it might be produced normally because of one of the following:

- An RRS cold start was performed that resulted in the loss of the RRS log. In this case, message DSN3010I should have been produced.
- RRS might have had a partial loss of log data during restart. In this case, message DSN3016I should have been produced.
- The RRS ISPF interface might have been used to heuristically commit or roll back the UR.

If any of the above are true, the message can be ignored. If not, there might be a DB2 or an RRS problem.

Problem determination: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 6, 33. 72.

DSN3018I *csect-name* **RRS HAS NO KNOWLEDGE OF DB2 URID=***db2_urid***. THE DB2 UR WAS COMMITTED. THE RRS URID RECORDED BY DB2=***rrs_urid***.**

Explanation: DB2 is restarting with RRS where DB2 is the coordinator and RRS is a participant. DB2 has a unit of recovery that RRS should be a participant in, but RRS has no knowledge of the DB2 unit of recovery. The DB2 unit of recovery ID is 'db2_urid'. DB2 has recorded the RRS unit of recovery ID as 'rrs_urid'.

System action: Restart with RRS continues.

Operator response: Notify the database administrator.

Database Administrator Action: This message might indicate a DB2 or RRS problem, or it might be produced normally because of one of the following:

- An RRS cold start was performed that resulted in the loss of the RRS log. In this case, message DSN3010I should have been produced.
- RRS might have had a partial loss of log data during restart. In this case, message DSN3016I should have been produced.
- The RRS ISPF interface might have been used to heuristically commit or roll back the UR.

If any of the above are true, the message can be ignored. If not, there might be a DB2 or an RRS problem.

Problem determination: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 6, 33. 72.

DSN3019I *csect-name* **RRS HAS NO KNOWLEDGE OF DB2 URID=***db2_urid***. THE DB2 UR WAS ROLLED BACK. THE RRS URID RECORDED BY DB2=***rrs_urid***.**

Explanation: DB2 is restarting with RRS where DB2 is the coordinator and RRS is a participant. DB2 has a unit of recovery that RRS should be a participant in, but RRS has no knowledge of the DB2 unit of recovery. The DB2 unit of recovery ID is 'db2_urid'. DB2 has recorded the RRS unit of recovery ID as 'rrs_urid'.

System action: Restart with RRS continues.

Operator response: Notify the database administrator.

Database Administrator Action: This message might indicate a DB2 or RRS problem, or it might be produced normally because of one of the following:

- An RRS cold start was performed that resulted in the loss of the RRS log. In this case, message DSN3010I should have been produced.
- RRS might have had a partial loss of log data during restart. In this case, message DSN3016I should have been produced.
- The RRS ISPF interface might have been used to heuristically commit or roll back the UR.

If any of the above are true, the message can be ignored. If not, there might be a DB2 or an RRS problem.

Problem determination: If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Collect the following diagnostic items listed in Appendix A, "Problem determination," on page 659: 1, 6, 33. 72.

DSN3020I *csect-name* **POSSIBLE HEURISTIC DAMAGE. RRS HAS COMMITTED THE UR WITH RRS URID=***rrs_urid***. THE DB2 UR WITH DB2 URID=***db2_urid* **IS INDOUBT.**

Explanation: DB2 is restarting with RRS and has a unit of recovery (UR) DB2 URID=db2_urid that is indoubt. DB2 is the commit coordinator for the UR and RRS is a participant. RRS has committed the same UR. The RRS URID=rrs_urid.

System action: Restart with RRS continues.

Operator response: Notify the database administrator.

Database Administrator Action: If possible, insure that the DB2 UR is resolved to commit. If this is done, there will be no heuristic damage. If the DB2 UR is resolved to abort, heuristic damage will have occurred since RRS committed the same UR.

DSN3021I *csect-name* **POSSIBLE HEURISTIC DAMAGE. RRS HAS ROLLED BACK THE UR WITH RRS. URID=***rrs_urid***. THE DB2 UR WITH DB2 URID=***db2_urid* **IS INDOUBT.**

Explanation: DB2 is restarting with RRS and has a unit of recovery (UR) DB2 URID=db2_urid that is indoubt. DB2 is the commit coordinator for the UR and RRS is a participant. RRS has rolled back the same UR. The RRS URID=rrs_urid.

System action: Restart with RRS continues.

Operator response: Notify the database administrator.

Database Administrator Action: If possible, insure that the DB2 UR is resolved to abort. If this is done, there will be no heuristic damage. If the DB2 UR is resolved to commit, heuristic damage will have occurred since RRS rolled back the same UR.

DSN3022I *csect-name* **HEURISTIC DAMAGE HAS OCCURRED. RRS HAS COMMITTED THE UR WITH RRS URID=***rrs_urid***, AND DB2 HAS ROLLED BACK THE SAME UR. THE DB2 URID=***db2_urid*

Explanation: DB2 is restarting with RRS and has a unit of recovery (UR) with DB2 URID=db2_urid that was rolled back. DB2 is the commit coordinator for the UR and RRS is a participant. RRS has committed the same UR. The RRS URID=rrs_urid.

System action: Restart with RRS continues.

Operator response: Notify the database administrator.

Database Administrator Action: Heuristic damage has occurred since DB2 rolled back the UR and RRS has committed the same UR. Correct the damage. This is not an easy task. Knowledge of the application is required to determine which Recoverable Resource Managers are involved in the UR. Since the time of the heuristic action, the data that was updated might have been used by many applications - read and/or write. Correcting the damage might involve reversing the effects of these applications as well. Tools available are:

- The RRS ISPF functions can provide assistance in determining the set of resource managers involved and what action they took.
- The DB2 DSN1LOGP utility provides a summary report that identifies the table spaces modified by the unit of recovery.

Contact your IBM Support Center for further assistance.

DSN3023I *csect-name* **HEURISTIC DAMAGE HAS OCCURRED. RRS HAS ROLLED BACK THE UR WITH RRS URID=***rrs_urid***, AND DB2 HAS COMMITTED THE SAME UR. THE DB2 URID=***db2_urid*

Explanation: DB2 is restarting with RRS and has a unit of recovery (UR) with DB2 URID=db2_urid that was committed. DB2 is the commit coordinator for the UR and RRS is a participant. RRS has rolled back the same UR. The RRS URID=rrs_urid.

System action: Restart with RRS continues.

Operator response: Notify the database administrator.

Database Administrator Action: Heuristic damage has occurred since DB2 committed the UR and RRS has rolled back the same UR. Correct the damage. This is not an easy task. Knowledge of the application is required to determine which Recoverable Resource Managers are involved in the UR. Since the time of the heuristic action, the data that was updated might have been used by many applications - read and/or write. Correcting the damage might involve reversing the

effects of these applications as well. Tools available are:

- The RRS ISPF functions can provide assistance in determining the set of resource managers involved and what action they took.
- The DB2 DSN1LOGP utility provides a summary report that identifies the table spaces modified by the unit of recovery.

Contact your IBM Support Center for further assistance.

DSN3024I *csect_name* **HEURISTIC DAMAGE HAS OCCURRED. COORDINATOR DB2 HAS COMMITTED THE UR WITH DB2 URID=***db2_urid***, AND RRS HAS INDICATED THAT THE OUTCOME WAS MIXED. THE RRS URID=***rrs_urid***.**

Explanation: DB2 is the commit coordinator and RRS is a participant. DB2 committed the UR with URID = db2_urid and RRS has indicated that one or more Recoverable Resource Managers that participated in the RRS UR with URID = rrs_urid has rolled back the same UR.

System action: Processing continues.

Operator response: Notify the database administrator.

Database Administrator Action: Heuristic damage has occurred since DB2 committed the UR and at least one RRS Recoverable Resource manager has rolled back the same UR. Correct the damage. This is not an easy task. Knowledge of the application is required to determine which Recoverable Resource Managers are involved in the UR. Since the time of the heuristic action, the data that was updated might have been used by many applications - read and/or write. Correcting the damage might involve reversing the effects of these applications as well. Tools available are:

- The RRS ISPF functions can provide assistance in determining the set of resource managers involved and what action they took.
- The DB2 DSN1LOGP utility provides a summary report that identifies the table spaces modified by the unit of recovery.

Contact your IBM Support Center for further assistance.

DSN3025I *csect-name* **HEURISTIC DAMAGE HAS OCCURRED. COORDINATOR DB2 HAS ROLLED BACK THE UR WITH DB2 URID=***db2_urid***, AND RRS HAS INDICATED THE OUTCOME WAS MIXED. THE RRS URID=***rrs_urid***.**

Explanation: DB2 is the commit coordinator and RRS is a participant. DB2 rolled back the UR with URID = db2_urid and RRS has indicated that one or more Recoverable Resource Managers that participated in the RRS UR with URID = rrs_urid has committed the same UR.

System action: Processing continues.

Operator response: Notify the database administrator.

Database Administrator Action: Heuristic damage has occurred since DB2 rolled back the UR and at least one RRS Recoverable Resource manager has committed the same UR. Correct the damage. This is not an easy task. Knowledge of the application is required to determine which Recoverable Resource Managers are involved in the UR. Since the time of the heuristic action, the data that was updated might have been used by many applications - read and/or write. Correcting the damage might involve reversing the effects of these applications as well. Tools available are:

- The RRS ISPF functions can provide assistance in determining the set of resource managers involved and what action they took.
- The DB2 DSN1LOGP utility provides a summary report that identifies the table spaces modified by the unit of recovery.

Contact your IBM Support Center for further assistance.

DSN3026I *csect-name* **POSSIBLE HEURISTIC DAMAGE. COORDINATOR DB2 HAS COMMITTED THE UR WITH DB2 URID=***db2_urid***, AND RRS HAS INDICATED THAT THE OUTCOME IS PENDING. THE RRS URID=***rrs_urid***.**

Explanation: DB2 is the commit coordinator for the DB2 UR with URID=db2_urid and RRS is a participant. RRS has also committed the UR but has indicated that at least one RRS Recoverable Resource manager has not completed commit processing for the RRS UR with URID=rrs_urid.

System action: Processing continues.

Operator response: Notify the database administrator.

Database Administrator Action: If possible, insure that the RRS Recoverable Resource Manager commits the RRS UR. If this is done, there will be no heuristic damage. If the RRS UR is rolled back by the RRS Recoverable Resource manager, heuristic damage will have occurred since DB2 committed the same UR.

DSN3027I *csect-name* **POSSIBLE HEURISTIC DAMAGE. COORDINATOR DB2 HAS ROLLED BACK THE UR WITH DB2 URID=***db2_urid***, AND RRS HAS INDICATED THAT THE OUTCOME IS PENDING. THE RRS URID=***rrs_urid***.**

Explanation: DB2 is the commit coordinator for the DB2 UR with URID=db2_urid and RRS is a participant. RRS has also rolled back the UR but has indicated that

at least one RRS Recoverable Resource manager has not completed roll back processing for the RRS UR with URID=rrs_urid

System action: Processing continues.

Operator response: Notify the database administrator.

Database Administrator Action: If possible, insure that the RRS Recoverable Resource Manager rolls back the RRS UR. If this is done, there will be no heuristic damage. If the RRS UR is committed by the RRS Recoverable Resource manager, heuristic damage will have occurred since DB2 rolled back the same UR.

DSN3028I *csect-name action-1* **CANNOT BE PERFORMED ON A UR BECAUSE THE UR IS IN A GLOBAL TRANSACTION, AND THE STATE OF THE GLOBAL TRANSACTION IS** *action-2*. *action-2* **IS PERFORMED. URID =** *rba* **CONNECTION NAME =** *connid* **CORRELATION ID =** *corrid* **LUWID =** *logical-unit-of-work-ID* **=** *token*

Explanation: A -RECOVER INDOUBT command was issued for a Unit of Recovery (UR) that was part of a global transaction. The recover action is inconsistent with the state of the global transaction. The requested recover action *action-1* is being changed to *action-2* to preserve the integrity of the global transaction. URID is the beginning RBA of the UR. The combination of Connection Name, Correlation ID, and LUWID identify the thread associated with the UR.

System action: The indoubt Unit of Recovery is resolved to *action-2*.

DSN3029I *csect-name* **RRS ATTACH PROCESSING IS AVAILABLE.**

Explanation: RRSAF has finished synchronizing with z/OS RRS and is now available to process application calls.

System action: Processing continues.

Operator response: Applications that require RRSAF services can be started.

DSN3100I *csect-name* **SUBSYSTEM** *ssnm* **READY FOR START COMMAND**

Explanation: The 'ssnm' in the message is the name of the subsystem that issued the message. This error occurs during MVS IPL processing. Message DSN3104I is also issued during this processing.

If AUTOSTART=YES is specified in DSNZPARM and this message is used as a trigger for automation to restart the DB2, a DELAY must be programmed into the automation to allow time for the IRLM to process the termination, and issue the STOP IRLM. Another approach would be to monitor for both the DB2 termination and the IRLM termination in combination, before restarting DB2. If neither of these actions are done, DB2 could identify to the existing IRLM and will NOT STOP automatically from that point onward.

Operator response: Issue the -START command when you are ready to start DB2.

DSN3101I *csect-name* **INVALID EARLY PROCESSING MODULE NAME IN PARMLIB RECORD.** *ssnm* **IS NOT AVAILABLE**

Explanation: CSECT DSN3UR00 has discovered that a parameter passed to DB2 subsystem initialization is in error. Either the Early processing load module name is not between 1 and 8 characters long, or there are syntax errors in the parameter string. The entire parameter string must be between 3 and 10 characters long, and the next to the last character must be a ','. The parameters were obtained from the parameter fields of a record defining a DB2 subsystem in member IEFSSNxx of SYS1.PARMLIB.

The subsystem recognition character is not inserted in this message as it is not yet known. 'ssnm' is the name of the DB2 subsystem undergoing IPL-time initialization.

System action: The DB2 subsystem with the indicated name is not available for this IPL of MVS.

Operator response: Notify the system programmer.

System programmer response: Correct the parameter fields in the record of SYS1.PARMLIB member IEFSSNxx, and re-IPL MVS. Refer to Part 2 of *DB2 Installation Guide* for the format of an IEFSSNxx record which defines a DB2 subsystem.

DSN3102I *csect-name* **INVALID SUBSYSTEM RECOGNITION CHARACTER IN PARMLIB RECORD.** *ssnm* **IS NOT AVAILABLE**

Explanation: CSECT DSN3UR00 detected that the subsystem recognition character is invalid; it must be nonblank. The subsystem recognition character is the last field of the parameter string passed to DB2 IPL initialization. It is obtained from a record for the named subsystem in member IEFSSNxx of SYS1.PARMLIB.

The subsystem recognition character is not inserted in this message as it cannot be determined. 'ssnm' is the name of the DB2 subsystem undergoing IPL-time initialization.

System action: The DB2 subsystem with the indicated name is not available for this IPL of MVS.

Operator response: Notify the system programmer. Correct the parameter fields in the SYS1.PARMLIB record of member IEFSSNxx, and re-IPL MVS. Refer to Part 2 of *DB2 Installation Guide* for the format of an IEFSSNxx record which defines a DB2 subsystem.

DSN3103I *csect-name* **- UNABLE TO LOAD EARLY PROCESSING MODULE SPECIFIED IN PARMLIB RECORD.** *ssnm* **IS NOT AVAILABLE**

Explanation: CSECT DSN3UR00 is unable to load the early processing load module named in the first parameter passed to DB2 subsystem IPL initialization. 'ssnm' is the name of the DB2 subsystem undergoing IPL-time initialization.

Either the load module name was specified incorrectly or the load module does not reside in a library included in the MVS linklist.

System action: DSN3UR00 abends with code X'04E' and reason code X'00F30103'. The DB2 subsystem with the indicated name is not available for this IPL of MVS.

Operator response: Notify the system programmer.

System programmer response: Identify the cause of the error. Either correct the early processing load module or take the necessary steps to include the module in a link-listed library. Re-IPL MVS. Refer to Part 2 of *DB2 Installation Guide* for the format of an IEFSSNxx record that defines a DB2 subsystem. Refer to the appropriate MVS publication for MVS linklist requirements.

Problem determination: An entry is made to SYS1.LOGREC. No SVC dump is taken.

DSN3104I *csect-name* **- TERMINATION COMPLETE**

Explanation: This message indicates that a DB2 subsystem has terminated. The actual MVS termination of the DB2 address space(s) may have completed earlier. This message is presented for every termination, normal or abnormal.

Operator response: Issue a DISPLAY ACTIVE command to see if MVS has completed termination processing of the address space(s).

DSN3105I *csect-name* **- UNABLE TO LOAD EARLY PROCESSING MODULE 'DSN3EPX'** *ssnm* **IS NOT AVAILABLE**

Explanation: CSECT DSN3UR00 is unable to load the early processing module DSN3EPX required for MVS. The name of the DB2 subsystem undergoing IPL-time initialization is 'ssnm'. Either the load module has been inadvertently deleted, or does not reside in a library included in the MVS linklist.

System action: DSN3UR00 abends with code X'04E' and reason code X'00F30105'. DB2 subsystem 'ssnm' is not available for this IPL of MVS.

Operator response: Notify the system programmer.

System programmer response: Identify the cause of the error. Either correct the early processing load module or take the necessary steps to include the module in a link-listed library. Re-IPL MVS.

If you suspect an error in DB2, refer to Part 2 of *DB2 Diagnosis Guide and Reference* for information on identifying and reporting the problem.

Problem determination: An entry is made in SYS1.LOGREC. No SVC dump is taken. Refer to MVS linklist requirements in *MVS/ESA JES2 Initialization and Tuning Guide*.

Collect the following diagnostic items listed in Appendix A, “Problem determination,” on page 659: 1, 5.

DSN3106I *csect-name* **SUBSYSTEM STOPPED. THIS COMMAND NOT PROCESSED:** *command-text*

Explanation: A command has been received which cannot be processed due to one of the following:

- The DB2 subsystem has not been started.
- The command was queued for processing while DB2 was starting, but startup terminated with an error.
- DB2 terminated before the command could be processed.

System action: The command is not processed.

Operator response: Start the DB2 subsystem, then reenter the command.

DSN3107I *csect-name* **COMMAND REJECTED. REQUESTER NOT AUTHORIZED**

Explanation: CSECT DSN3EC0X has discovered that the source of the command was a console that does not have 'system' authority.

System action: The command is not processed. This message is sent to the console that entered the command.

Operator response: Enter the command from another console that has 'system' authority.

System programmer response: Verify that this console should be used for entering DB2 commands. If so, authorize it for 'system' commands by using MVS services.

DSN3108I *csect-name* **COMMAND REJECTED. COMMAND FACILITY PATH UNAVAILABLE**

Explanation: CSECT DSN3EC0X has discovered that the path from MVS consoles to the DB2 command processor is unavailable. It may still be possible to enter commands through one of the attachment facilities.

System action: The command is not processed. This

message is delivered to the console that entered the command.

System programmer response: The console command facility is available again the next time DB2 is started. If it is necessary to regain this facility immediately, DB2 should be canceled and restarted.

DSN3109I *csect-name* **- UNABLE TO OBTAIN SUBSYSTEM AFFINITY TABLE INDEX FOR SUBSYSTEM** *ssnm***. IEFSSREQ RC =** *nn*

Explanation: CSECT DSN3UR00 was unable to obtain a subsystem affinity table index for the named subsystem. MVS did not recognize the named subsystem name as a known subsystem. If this message is issued, a serious error has occurred in MVS or DB2.

In the message, 'nn' is the return code from the IEFSSREQ MVS service. 'ssnm' is the name of the DB2 subsystem undergoing IPL-time initialization.

System action: DSN3UR00 abends with completion code X'04E' and reason code X'00F30104'. The DB2 subsystem with the indicated name is not available for this IPL of MVS.

Operator response: Notify the system programmer.

System programmer response: Try to re-IPL. If the problem persists, follow standard DB2 problem reporting procedures.

Problem determination: A record is written to SYS1.LOGREC. No SVC dump is taken. Return codes from IEFSSREQ are documented under control block SSOB in the appropriate MVS publication.

DSN3111I *csect-name* **LOAD MODULE** *module-name* **DOES NOT HAVE AMODE(31) ATTRIBUTE.** *ssnm* **IS NOT AVAILABLE**

Explanation: The DB2 program which establishes DB2 as an MVS subsystem during Master Scheduler initialization has determined that either its own load module or a load module loaded by it does not have the linkage editor attributes AMODE(31) and RMODE(ANY). Most DB2 load modules must be capable of running in the 31-bit addressing mode.

In the message, 'csect-name' is the name of the program detecting the error, 'module-name' is the name of the first load module found with an invalid AMODE attribute, and 'ssnm' is the name of the DB2 subsystem undergoing IPL-time initialization.

System action: Program DSN3UR00 abends with a code of X'04E' and a reason code of X'00F30111'. The DB2 subsystem named 'ssnm' is not initialized. The subsystem will not be available until the error is corrected and MVS is re-IPLed.

Operator response: Notify the system programmer of the failure.

System programmer response: Verify that all of the following were done:

- The installation and maintenance activities against DB2 were executed using SMP.
- The JCLIN for SMP included the correct AMODE and RMODE control statements.
- The appropriate Linkage Editor provided with the Data Facility Product (DFP) for the version of MVS being used was invoked by SMP for the link-edit steps.

Correct and rerun the procedure or JCL which caused the error. If necessary, REJECT all SMP APPLY steps which were executed incorrectly and rerun the rejected steps. Request a re-IPL of MVS.

Problem determination: A record is written to SYS1.LOGREC. No SVC dump is taken.

The MVS service aid AMBLIST provides Linkage Editor module attributes in the summary section of the LISTLOAD function output.

DSN3112I *csect-name* **UNABLE TO REGISTER COMMAND PREFIX, CPF RETURN CODE =** *xxyy***.** *ssnm* **IS NOT AVAILABLE**

Explanation: DB2 is unable to register the installation-specified command prefix with MVS. The command prefix is specified in the subsystem definition parameter. *xx* is the hexadecimal return code, and *yy* is the hexadecimal reason code from the CPF MVS service. Subsystem *ssnm* is not available.

System action: This DB2 subsystem is not available.

System programmer response: Refer to the MVS documentation on the CPF macro service to determine the reason for the *xx* and *yy* codes. Ensure that the specified command prefix is not already registered, or is not a substring or superstring of an already registered command prefix. Use the MVS DISPLAY OPDATA command, if necessary, to see command prefixes registered with MVS.

DSN3113I *csect-name* **INVALID SUBSYSTEM DEFINITION PARAMETER IN PARMLIB RECORD.** *ssnm* **IS NOT AVAILABLE.**

Explanation: DB2 detected an error in the subsystem definition parameter, which is obtained from the IEFSSNxx member of SYS1.PARMLIB. Some possible causes for this error are:

- The early processing load module name is not between 1 and 8 characters long.
- The command prefix is not between 1 and 8 characters long.
- The command prefix scope is not 1 character long.
- The command prefix scope is not X, S, or M.

- The group attach name is not between 1 and 4 characters long.

The command prefix is not inserted into this message because it is unknown. The name of the subsystem attempting to initialize is *ssnm*.

System action: The DB2 subsystem is not available.

System programmer response: Correct the subsystem definition parameter and re-IPL MVS. Refer to *DB2 Installation Guide* for the format of the subsystem definition parameter that defines a DB2 subsystem.

DSN3114I *csect-name* **GROUP ATTACH NAME IS NOT ALLOWED WITH THIS RELEASE OF MVS.** *ssnm* **IS NOT AVAILABLE**

Explanation: DB2 found an installation-specified group attachment name specified in the subsystem definition parameter, IEFSSNxx, but DB2 group attachment is not supported with this release of MVS. DB2's group attachment feature requires MVS/ESA SP4.2.2 or a later release. The DB2 subsystem *ssnm* is not available.

System programmer response: Delete the group attachment name in the subsystem definition parameter and re-IPL.

DSN3115I *csect-name* **UNABLE TO REGISTER GROUP ATTACH NAME, NAME/TOKEN RC =** *xx***.** *ssnm* **IS NOT AVAILABLE**

Explanation: DB2 is unable to register the installation-specified group attach name with MVS. The group attach name is specified in the subsystem definition parameter. *xx* is the hexadecimal return code from the MVS Name/Token service. Subsystem *ssnm* is not available.

System action: This DB2 subsystem is not available.

System programmer response: Refer to MVS documentation about the Name/Token macro service to determine the reason for the *xx* return codes.

DSN3116I *csect-name* **THE VERSION OF THE DB2 EARLY CODE IS INCOMPATIBLE WITH THIS VERSION OF MVS.** *ssnm* **IS NOT AVAILABLE**

Explanation: This version of DB2 early code requires MVS/ESA 4.2.0. Subsystem *ssnm* is not available.

System programmer response: The version of the DB2 early code that was loaded from the LINKLIST library is not compatible with the release of MVS that you are using. Change the concatenation order of the LINKLIST library so that a compatible release of DB2 early code is used.

DSN3201I **ABNORMAL EOT IN PROGRESS FOR USER=***xxxxxxxx* **CONNECTION-ID=***yyyyyyyy* **CORRELATION-ID=***zzzzzzzzzzzz* **JOBNAME=***aaaaaaaa* **ASID=***9999* **TCB=***99999999*

Explanation: Abnormal termination started for the agent with the values shown for USER, CONNECTION-ID, CORRELATION-ID, JOBNAME, ASID and TCB address. These values are the last known set of identifiers for the terminating agent.

The abnormal termination could be the result of a failure in the allied agent's address space or the result of an operator cancel command.

The value for the USER and/or CORRELATION-ID might be blank. If the terminating agent is connected to DB2 at the IDENTIFY level and was not previously signed on, or was not previously authorized to a plan at the time of the failure, then no CORRELATION-ID is associated with the agent. Otherwise, the values for the USER, CONNECTION-ID, CORRELATION-ID, JOBNAME and ASID are the last values established to DB2 for this connection. They might represent the current activity of the agent. The TCB value is the address of the TCB that is terminating. Previous DB2 work by this agent might have completed successfully. Correlation IDs beginning with '0nn', where nn is a number from 01 to 28, identify system agents.

System action: The agent was previously queued to a System Services service task for termination processing. This message indicates that the agent was taken from the queue for processing. Any uncommitted database changes are rolled back.

User response: Your job or TSO session cannot be canceled. The PA1 key has no effect in this circumstance. To avoid long waits when your application program is canceled or terminates abnormally, issue frequent commits.

System programmer response: See the Problem Determination section for this message. The MVS CANCEL and FORCE commands have no effect and should not be issued. DB2 should not be canceled. If an extensive rollback is in progress, the subsequent DB2 restart might take a very long time because of additional log activity.

Problem determination: You can detect a deferred termination condition for a task by examining several indicators. Some or all of the following might be present:

- The allied address space might be swapped out and appear to be in a never-ending WAIT condition.
- The MVS commands CANCEL and FORCE appear to have no effect.
- The allied task holds an MVS-shared ENQ on resource SYSZDSN3.ERLYOLRHerly-block-address.
- During abnormal termination of the agent associated with the failing task, but before this message was written, the task's connection appears on the

-DISPLAY THREAD output with a QD status. See message DSNV404I for the definition of this status code.
- This message, DSN3201I, is written to the MVS console after the agent is removed from the service task work queue when termination processing begins.
- During abnormal termination of the agent associated with the failing task, the task's connection appears on the -DISPLAY THREAD output with a D status. See message DSNV404I for the definition of this status code.

DSN3458I *csect-name* **MODULE** *modname* **MUST HAVE THE AMODE (31) ATTRIBUTE**

Explanation: An authorization exit load module was not link-edited with the 31-bit addressing mode (AMODE) attribute. Either load module DSN3@ATH or DSN3@SGN is specified by 'modname'.

System action: Subsystem startup is terminated.

System programmer response: Relink-edit the authorization exit module specifying the attributes AMODE(31), RMODE(ANY).

Problem determination: Check the load module attributes used for the link-edit.

DSN3580I **CONNECTION FOR [EOM | FEOT | IDEN] GAVE RC=X xxxxxxxx, REASON=X xxxxxxxx**

Explanation: A nonzero return code has been returned to DSN3AMI2 from the SSSS connect to subsystem interface call. The variables in the message indicate which SSI call is involved and the actual return and reason codes associated with it.

System action: The current task (TCB) is abended with a system completion code of X'04E' and with an abend reason code of '00F30580' The DB2 subsystem terminates.

Operator response: Notify the system programmer.

System programmer response: Restart the DB2 subsystem.

Problem determination: The trace table and stack storage associated with the EB in register 6 must be examined to locate the error.

Collect the following diagnostic items listed in Appendix A, “Problem determination,” on page 659: 2, 5.

DSN3581I *csect-name* **SAF** *service* **FAILED WITH SAF RETURN CODE =** *saf-retc*, **RACF RETURN CODE =** *racf-retc*, **AND RACF REASON CODE =** *racf-reac*. **DB2 REASON CODE =** *db2-reac*

Explanation: DB2 invoked the specified SAF (Security Authorization Facility) service which failed with the following codes:

Variable	Definition
csect	The CSECT name of the DB2 module that issued the message.
service	The name of the service that failed.
saf-retc	The SAF return code.
racf-retc	The RACF return code.
db2-retc	The DB2-assigned reason code for the failure. DB2 associates the failure with a DB2 reason code which may appear in other messages associated with this failure.

System action: The call fails. Other messages associated with this feature may follow.

User response: Notify the system programmer.

Operator response: Notify the system programmer.

System programmer response: The action to take depends on the service being used.

- R_ticketserv

  DB2 requests this service to parse a Kerberos ticket received from a remote client. Refer to OS/390 Security Server (RACF) Callable services for an exlanation of the error.

  DB2 uses the SAF R_ticketserv service to parse a Kerberos ticket that was received from a remote client. Refer to *OS/390 SecureWay Security Server RACF Callable Services* for an explanation of the error. A SAF and RACF return code of 8 and 16 indicate a failure from the Network Authentication Privacy Service (Kerberos) when parsing the ticket in which case the RACF reason code contains a status code from the Network Authentication Privacy Service. These status codes are documented in the *OS/390 SecureWaySecurity Server Network Authentication and Privacy Service Administration* manual. Additional information for some of the more common status codes and how they relate to DB2 follows:

  96C73A07
  One possibility is that you must associate the ssnmDIST started task user ID with a principal name and the client system must use this principal name in the Kerberos target principal.

  Note: DB2 Connect client systems must catalog a DB2 for OS/390 database with "AUTHENTICATION KERBEROS TARGET PRINCIPAL principal_name". Assume that SYSDSP is the user ID associated

with the ssnmDIST started task. A principal, say SYSDSPp, must be associated to the SYSDSP user ID by a RACF ALTUSER command. Further, the SYSDSPp principal name must appear in the target principal name of the DB2 Connect database configuration for the DB2 for OS/390 database.

- R_ticketserv(principal name)

  DB2 uses the SAF R_usermap service to determine the user ID associated with a Kerberos ticket received from a remote client. Refer to *OS/390 SecureWay Security Server RACF Callable Services* for an explanation of the error. The principal name obtained from the ticket is included in parenthesis to assist the user in determining the cause of the problem. Either the client system is using an invalid principal name, or a principal name to user ID association has not been defined to RACF.

  Principals authenticated on the local OS/390 realm must have a mapping that associates the local principal name to a local RACF user ID. This association can be made by using the RACF ADDUSER or ALTUSER commands with the KERBNAME parameter.

  Principals authenticated on a foreign realm must have a mapping that associates the foreign fully qualified principal name to a local RACF user ID. This association can be made by using the RACF RDEFINE or RALTER commands with the KERBLINK parameter.

- R_usermap

  DB2 requests this service to determine the user ID associated with a Kerberos ticket received from a remote client. Refer to *OS/390 Security Server (RACF) Callable services* for an explanation of the error.

Chapter 27. DSN7... messages

DSN7000I *csect-name* **REQUIRED CSECT DSN6GRP COULD NOT BE FOUND IN SYSTEM PARAMETERS LOAD MODULE.**

Explanation: The required CSECT DSN6GRP was not found in the system parameters load module.

System action: The DB2 subsystem terminates with reason code 00F70001.

Operator response: Restart DB2 with the correct system parameters load module.

System programmer response: Reassemble the system parameters load module and include the DSN6GRP macro.

DSN7001I *csect-name* **DATA SHARING SPECIFIED IN SYSTEM PARAMETERS, BUT NOT RUNNING ON A MVS SYSTEM LEVEL WHICH IS REQUIRED FOR DATA SHARING.**

Explanation: A DB2 subsystem was started with the data sharing option, but MVS is not at the required level. DB2 data sharing requires MVS Version 5 Release 1.0 or a later release.

System action: The DB2 subsystem terminates with reason code 00F70002.

Operator response: Restart DB2 on MVS Version 5 Release 1.0 or a later release.

DSN7002I **LIGHT(YES) ON START DB2 COMMAND WAS IGNORED, SYSTEM IS NOT ENABLED FOR DATA SHARING**

Explanation: DB2 issues this message during DB2 startup if DB2 is started with the LIGHT(YES) keyword and the system is not enabled for data sharing.

System action: DB2 startup continues normally. DB2 ignores the LIGHT(YES) specification.

DSN7100I *csect-name*

Explanation: This message indicates the beginning of the output from a DISPLAY GROUP command. It is the first line of a multi-line message response.

DSN7101I ***** BEGIN DISPLAY OF GROUP(*group-name*) GROUP LEVEL (*group-level*) MODE (*nn*)**

Explanation: This message is a line of output from a DISPLAY GROUP command.

group-name
: The name of the data sharing group.

group-level
: A string of three numeric characters:
- Version
- Release
- Modification level

nn Two characters that indicate the mode of the system and can have the following possible values:

C	Compatibility mode
E	Enabling new function mode
N	New function mode
C*	Compatibility mode but the system detects that it has been to a higher level
E*	Enabling new function mode but the system detects that it has been to a higher level

DSN7102I **----------**

Explanation: This message is a line of output from a DISPLAY GROUP command.

DSN7103I **DB2 DB2 SYSTEM IRLM**

Explanation: This message is a line of output from a DISPLAY GROUP command.

DSN7104I **MEMBER ID SUBSYS CMDPREF STATUS LVL NAME SUBSYS IRLMPROC**

Explanation: This message is a line of output from a DISPLAY GROUP command.

MEMBER
: Name of the DB2 data sharing member.

ID
: Internally assigned ID for each data sharing member.

SUBSYS
: Subsystem name of the data sharing member.

CMDPREF
: Command prefix of the data sharing member.

STATUS
: State of the data sharing member, which can

© Copyright IBM Corp. 1983, 2008

be 'ACTIVE', 'AI', 'FAILED', 'QUIESCED', 'QI', 'QR', 'QC', 'QIC', 'QIR', or 'QICR' where:

ACTIVE
The member is currently up and running.

AI The member is active with additional conditions. Indoubt or postponed abort units of recovery are outstanding. This means retained locks are held.

FAILED
The member has terminated abnormally.

QUIESCED
The member is down normally.

QI The member is down with indoubt URs.

QR The member is down with resynchronization responsibility.

QC The member is down with group buffer pool castout responsibility.

QCR, QIC,QIR,QICR
A possible combination of the quiesced states that are listed above.

LVL String of three numeric characters as follows:
- DB2 Version
- DB2 Release
- DB2 Modification level

SYSTEM NAME
Name of the MVS system on which the data sharing member runs.

IRLM SUBSYS
The accompanied IRLM's subsystem name for this DB2.

IRLMPROC
The accompanied IRLM's startup procedure name.

DSN7105I **-------- --- ---- -------- -------- --- -------- ---- --------**

Explanation: This message is a line of output from a DISPLAY GROUP command.

DSN7106I *member-name subsystem-name command-prefix member-status mvs-system-name irlm-subsystem-name irlm-procedure-name*

Explanation: This message is a line of output from a DISPLAY GROUP command. *member-status* can be ACTIVE, QUIESCED, or FAILED. *member-status* may also have the string '***' as output if the member has not been quiesed normally.

QUIESCED
This is a normal quiesced state, as the result of a normal STOP DB2 command.

Q The member is quiesced, but with additional conditions. Q (quiesced) can be paired with one or more of the following letters:

I Indoubt units of recovery (URs) are outstanding. This means retained locks are held.

C There was a castout error. The last updated of the page set or partition could not write from the coupling facility to DASD.

Make sure there is connectivity from the coupling facility to DASD before restarting DB2.

R There is retained information needed for DB2 to perform resynchronization with one or more remote locations.

When DB2 is restarted, this resynchronization occurs.

ACTIVE
This is a normal active state without conditions

A The member is active, but with the addtional conditions. A (active) can be paired with the following letter:

I Indoubt or postponed abort units of recovery (URs) are outstanding. This means retained locks are held.

*** Usually issued if the member has not been quiesced normally. The string '***' may also be paired with the following:

**** Information unknown. This may result if an unwanted member was previously added to the group, and continues to be named in other locations.

The *subsystem-name*, *command-prefix*, *irlm-subsystem-name*, and *irlm-procedure-name* show '****' if this information is unknown.

DSN7107I **PROTOCOL LEVEL(*protocol-level*)**
GROUP ATTACH NAME(*group-attach*)

Explanation: This message is a line of output from a DISPLAY GROUP command.

group-attach is the group attach name for the data sharing group.

protocol-level indicates the current locking protocol used
in the data sharing group.

DSN7108I NUMBER LOCK ENTRIES: *nnnnnnnnnnn*

Explanation: This line of output from a DISPLAY GROUP command shows information about the LOCK1 coupling facility structure. The maximum number of lock table entries is displayed.

DSN7109I SCA STRUCTURE SIZE: *nnnnnnn* **KB, STATUS =** *xx*, **SCA IN USE:** < *nnn* **%**

Explanation: This message is a line of output from a DISPLAY GROUP command.

If the % value is > 0 but < 1, then a ″<″ character is shown before the 1 % value, e.g. ″< 1 %″. This can happen if the shared communications area (SCA) structure is very large or if only a small number of SCA entries are in use.

The SCA status can have following values:

IN SCA is in initialization phase.

AC SCA is active.

DS SCA loss of connectivity, delayed action started.

DC SCA loss of connectivity, delayed action completed.

QS SCA rebuild quiesce function started.

QC SCA rebuild quiesce function completed.

CS SCA rebuild connect function started.

CC SCA rebuild connect function completed.

OS SCA rebuild connect complete function started.

RS SCA rebuild resource manager invocation started.

RC SCA rebuild resource manager invocation completed.

OC SCA rebuild connect complete function completed.

LS SCA rebuild cleanup function started.

US SCA rebuild resource manager invocation started.

UC SCA rebuild resource manager invocation completed.

LC SCA rebuild cleanup function completed.

PS SCA rebuild process complete function started.

PC SCA rebuild process complete function completed.

SS SCA rebuild stop function started.

SC SCA rebuild stop function completed.

ES SCA rebuild stop complete function started.

EC SCA rebuild stop complete function completed.

VS SCA volatility change function started.

VC SCA volatility change function completed.

TE SCA is in termination phase.

DSN7110I LOCK1 STRUCTURE SIZE: *nnnnnnn* **KB**

Explanation: This line of output from a DISPLAY GROUP command shows the size of the LOCK1 coupling facility structure.

DSN7111I NUMBER LIST ENTRIES: *nnnnnnnnnnn*, **LIST ENTRIES IN USE:** < *nnnnnnnnnnn*

Explanation: This line of output from a DISPLAY GROUP command shows information about the LOCK1 coupling facility structure. The maximum number of modify lock list entries and the number of those entries currently in use are displayed. (MVS documentation refers to these list entries as “record data”.)

DSN7112I * END DISPLAY OF GROUP(***group-name***)**

Explanation: This message is the last line of output from a DISPLAY GROUP command.

DSN7113I DISPLAY TERMINATED DUE TO INSUFFICIENT MESSAGE SPACE.

Explanation: There is not enough storage available to store the required information from this command.

System action: The DB2 subsystem displays the group information, but some information is missing.

DSN7114I XCF OR STRUCTURE INFORMATION NOT AVAILABLE. MVS *nnnnnnnn* **RETURN CODE=***xxxxxxxx*, **MVS** *nnnnnnnn* **REASON CODE=***yyyyyyyy*.

Explanation: An error occurred during the query of the cross-system coupling facility (XCF) or CF structure information.

nnnnnnnn could be IXCQUERY or IXLMG.

System action: The DB2 subsystem displays the group, but some XCF or coupling facility structure information is missing. Refer to *MVS/ESA Programming: Sysplex Services Reference* for an explanation of IXCQUERY or IXLMG return and reason codes.

DSN7120I DB2 PARALLEL PARALLEL

Explanation: This message is a line of output from a DISPLAY GROUP command with the DETAIL option.

DSN7121I MEMBER COORDINATOR ASSISTANT

Explanation: This message is a line of output from a DISPLAY GROUP command with the DETAIL option.

DSN7122I -------- ---------- ---------

Explanation: This message is a line of output from a DISPLAY GROUP command with the DETAIL option.

DSN7123I *member-name parallel-coordinator parallel-assistant*

Explanation: This message is a line of output from a DISPLAY GROUP command with the DETAIL option. This message is only issued for ACTIVE DB2 members that support Sysplex query parallelism.

DSN7130I TABLE ENABLED FOR

Explanation: This message is a line of output from a
DISPLAY GROUP command with the DETAIL option.
This message is only issued when the DB2 catalog
mode is 'E' as indicated on the Display Group
DSN7101I message output

DSN7131I SPACE NEW FUNCTION

Explanation: This message is a line of output from a
DISPLAY GROUP command with the DETAIL option.
This message is only issued when the DB2 catalog
mode is 'E' as indicated on the Display Group
DSN7101I message output

DSN7132I -------- ----------

Explanation: This message is a line of output from a
DISPLAY GROUP command with the DETAIL option.
This message is only issued when the DB2 catalog
mode is 'E' as indicated on the Display Group
DSN7101I message output

DSN7133I *tablespace-name enable-newfunction*

Explanation: This message is a line of output from a
DISPLAY GROUP command with the DETAIL option.
This message is only issued when the DB2 catalog
mode is 'E' as indicated on the Display Group
DSN7101I message output

enable-newfunction indicates whether this table space has
been converted to UNICODE during the enabling new
function mode process.

DSN7401I *csect-name* **INCONSISTENT DATA SHARING PARAMETER DETECTED,** *parameter-name* **IN DB2 SYSTEM PARAMETERS (***sysparm-parm***) IS NOT EQUAL TO PARAMETER IN THE DB2 START-UP PROCEDURE (***start-parm***).**

Explanation: At least one data sharing parameter in the member's system parameters is not equal to the data sharing parameter value passed on the DB2 start-up procedure.

System action: The DB2 subsystem terminates with reason code 00F70003.

Operator response: Restart DB2 with the correct system parameters load module or inform the system programmer to change the DB2 start-up procedure.

System programmer response: Reassemble the system parameters load module. Make sure that the group name and the member name match the values in the DB2 start-up procedure, or correct the parameters on the DB2 start-up procedure.

DSN7403I *csect-name* **MEMBER(***member-name***) IS ALREADY ACTIVE ON SYSTEM(***mvs-system-name***).**

Explanation: The started DB2 subsystem is already active on another MVS system.

System action: The DB2 subsystem terminates with reason code 00F70003.

Operator response: If you want the DB2 subsystem to run on this MVS system, stop the DB2 subsystem on the other MVS. After the DB2 subsystem stops, restart the DB2 on this MVS.

DSN7404I *csect-name* **INVALID MEMBER(***member-name***) WAS FOUND IN DB2 XCF GROUP(***db2-group-name***). MEMBER HAS BEEN DELETED.**

Explanation: The invalid member was found in a DB2 data sharing group, which is a particular type of cross-system coupling facility (XCF) group. The invalid name *member-name* was found in data sharing group *db2-group-name* and was deleted. *member-name* might represent an unauthorized program, other than DB2 for MVS/ESA, that attempted to join the data sharing group.

System action: The DB2 subsystem continues.

Operator response: Report the occurrence of this message to your system programmer.

System programmer response: Investigate which unauthorized program joined the DB2 group and find its owner. This could indicate a security violation.

DSN7405I *csect-name* **MEMBER(***member-name***) UNABLE TO JOIN DATA SHARING GROUP(***group-name***), MVS IXCJOIN RETURN CODE=***xxxxxxxx***, MVS IXCJOIN REASON CODE=***yyyyyyyy***.**

Explanation: An unexpected error occurred when the data sharing DB2 member attempted to join the MVS cross-system coupling facility (XCF) data sharing group.

System action: The DB2 subsystem terminates with reason code 00F70408.

Operator response: Issue an MVS 'DISPLAY XCF,GROUP' command to ensure that XCF is functional; then try to restart the DB2 subsystem.

System programmer response: Refer to *MVS/ESA Programming: Sysplex Services Reference* for an explanation of the IXCJOIN return and reason codes.

DSN7406I *csect-name* **UNABLE TO QUERY DATA SHARING GROUP INFORMATION, MVS IXCQUERY RETURN CODE=***xxxxxxxx***, MVS IXCQUERY REASON CODE=***yyyyyyyy***.**

Explanation: An unexpected error occurred during the execution of the MVS IXCQUERY macro.

System action: The DB2 subsystem terminates with reason code 00F70406.

Operator response: Issue an MVS DISPLAY XCF,GROUP command to ensure that XCF is functional. Then try to restart the DB2 subsystem again.

System programmer response: Refer to *MVS/ESA Programming: Sysplex Services Reference* for an explanation of the IXCQUERY return and reason codes.

DSN7407I *csect-name* **AUTOMATIC RESTART MANAGER** *function* **FAILED. MVS IXCARM RETURN CODE =** *xxxxxxxx***, MVS IXCARM REASON CODE =** *yyyyyyyy***.**

Explanation: DB2 encountered an error using the specified MVS automatic restart manager (ARM) function.

System action: DB2 continues. However, if DB2 abnormally terminates, the MVS ARM cannot automatically restart DB2.

System programmer response: Refer to *MVS/ESA Programming: Sysplex Services Reference* for the function, return code, and reason code from the IXCARM request.

DSN7408I *csect-name* **ARM EVENT EXIT FAILED. DB2 SUBSYSTEM** *ssnm* **NOT RESTARTED. MVS IEFSSI RETURN CODE =** *xxxxxxxx***, MVS IEFSSI REASON CODE =** *yyyyyyyy***.**

Explanation: The DB2 ARM event exit encountered an error while attempting to define the DB2 subsystem to MVS.

System action: The automatic restart of the DB2 subsystem is stopped.

System programmer response: Refer to *MVS/ESA Programming: Assembler Services Reference* for return and reason codes from the IEFSSI request. If the IEFSSI return code and reason code are 0, look in the console log for previous messages that indicate why the definition of the DB2 subsystem failed. You can manually start DB2 by using the -START DB2 command on any system where the DB2 subsystem is defined.

DSN7409I *csect-name* **MEMBER(***member-name***) UNABLE TO QUIESCE FROM DATA SHARING GROUP(***group-name***), MVS IXCQUIES RETURN CODE=***xxxxxxxx***, MVS IXCQUIES REASON CODE=***yyyyyyyy***.**

Explanation: An unexpected error occurred when the data sharing DB2 member attempted to place itself in an MVS cross-system coupling facility (XCF) quiesced state.

System action: The DB2 subsystem termination continues.

Operator response: Inform your system programmer.

System programmer response: Refer to *MVS/ESA Programming: Sysplex Services Reference* for an explanation of the IXCQUIES return and reason codes.

DSN7501A *csect-name* **SCA STRUCTURE** *sca-structure-name* **CONNECTIVITY FAILURE.**

Explanation: The specified DB2 member lost connectivity to the shared communication area (SCA) structure.

System action: The specified DB2 disconnects from the SCA and terminates with reason code 00F70600.

Operator response: Notify the system programmer.

System programmer response: Find out why the link from this central processor complex (CPC) to the SCA structure was lost. After fixing it, restart DB2.

DSN7502I *csect-name* **SCA STRUCTURE** *sca-structure-name* **FAILURE, ATTEMPT TO REBUILD IS IN PROGRESS.**

Explanation: The specified DB2 member detected the shared communications area (SCA) structure failure.

System action: DB2 tries to rebuild the SCA structure on a coupling facility in the preference list for the coupling facility.

Operator response: Wait for message DSN7503I or DSN7504A.

DSN7503I *csect-name* **SCA STRUCTURE** *sca-structure-name* **REBUILD SUCCESSFUL.**

Explanation: This member successfully connected to a shared communications area (SCA) structure on the same or an alternate coupling facility and completed restoration of its information in the shared communications area (SCA).

System action: DB2 processing continues.

DSN7504I *csect-name* **SCA STRUCTURE** *sca-structure-name* **REBUILD UNSUCCESSFUL. REASON CODE =** *xx***.**

Explanation: This member has not successfully connected to a shared communication area (SCA) structure on an alternate coupling facility and completed restoration of its information in the shared communications area (SCA). This message follows message DSN7502I.

04 An operator requested to stop the rebuild process.

08 A DB2 resource manager requested to stop the rebuild process.

12 The new SCA structure failed during the rebuild process.

16 A DB2 member lost connectivity to the new SCA in the rebuild process.

20 An MVS service used during the rebuild process failed. A message was issued from the failed MVS service.

System action: This DB2 member terminates with reason code 00F70601. Other members can continue processing.

System programmer response: See the termination code for more details.

DSN7505A *csect-name* **THERE IS A SHORTAGE OF FREE STORAGE IN SCA STRUCTURE** *sca-structure-name*

Explanation: The amount of free storage in the shared communications area (SCA) coupling facility structure is low. The SCA is 75 percent full. Because SCA storage space is required by some DB2 commands and processes, the availability of those processes might be affected. Completely running out of SCA storage can cause DB2 subsystem outages.

This message stays on the MVS console and is highlighted.

System action: DB2 processing continues.

This message might remain highlighted and might not be automatically deleted in a pre SP5.2.1-JES3 environment.

Operator response: If the message remains highlighted after the error situation is cleared up, manually delete the message from the console.

System programmer response: You can monitor the amount of SCA storage in use with the DISPLAY GROUP command.

You can free SCA storage by removing exception conditions. Another thing you can do is issue START DATABASE commands with the SPACENAM option or use the RECOVER utility to remove pages from the logical page list (LPL).

If you don't do anything to reclaim space, message DSN7512A appears when the SCA is 90 percent full.

DSN7506A *csect-name* **CONNECT TO THE SCA STRUCTURE** *sca-structure-name* **FAILED. MVS IXLCONN RETURN CODE=***xxxxxxxx***, MVS IXLCONN REASON CODE=***yyyyyyyy***.**

Explanation: DB2 cannot connect to the shared communications area (SCA) coupling facility structure.

System action: DB2 startup terminates with reason code 00F70602.

System programmer response: DB2 uses the MVS IXLCONN service to connect to the SCA structure. Refer to *MVS/ESA Programming: Sysplex Services Reference* for return and reason codes from the IXLCONN request.

DSN7507I *csect-name* **SCA STRUCTURE** *sca-structure-name* **IS ALLOCATED IN A VOLATILE STRUCTURE.**

Explanation: The shared communications area (SCA) structure is volatile.

System action: DB2 processing continues.

System programmer response: One DB2 issues this

message whenever the structure has the volatile attribute. DB2 detects this either on the first connection to the SCA or on the attribute changes later. DB2 always requests a nonvolatile SCA. If this is not what you want, change the MVS CFRM administrative policy and request a rebuild of the SCA with the MVS SETXCF START,REBUILD command.

DSN7508I *csect-name* **ACCESS TO THE SCA STRUCTURE** *sca-structure-name* **FAILED. MVS IXLLIST RETURN CODE =** *xxxxxxxx*, **MVS IXLLIST REASON CODE =** *yyyyyyyy*.

Explanation: DB2 encountered an error in accessing the shared communications area (SCA) structure.

System action: DB2 terminates with reason code 00F70603.

System programmer response: DB2 uses the MVS IXLLIST service to access the SCA structure. Refer to *MVS/ESA Programming: Sysplex Services Reference* for return and reason codes from the IXLLIST request.

DSN7509I *csect-name* **SCA STRUCTURE** *sca-structure-name* **IS ALLOCATED IN A NONVOLATILE STRUCTURE.**

Explanation: The SCA structure is now contained in a nonvolatile coupling facility structure.

System action: DB2 processing continues.

DSN7510I *csect-name* **SCA STRUCTURE** *sca-structure-name* **REBUILD REQUEST RECEIVED, BUT SUBSYSTEM IS NOT READY TO PARTICIPATE.**

Explanation: The DB2 subsystem received a request from MVS to rebuild the shared communications area (SCA), but DB2 is not ready to participate in this process because DB2 startup has not yet completed.

System action: DB2 terminates with reason code 00F70604. The SCA rebuild process might continue for other DB2s in the data sharing group.

Operator response: Wait until the SCA rebuild process is finished and then start the failed DB2 again.

DSN7511I *csect-name* **RESPONSE TO MVS REBUILD COMPLETE OF SCA STRUCTURE** *sca-structure-name* **FAILED. MVS IXLREBLD RETURN CODE =** *xxxxxxxx*, **MVS IXLREBLD REASON CODE =** *yyyyyyyy*.

Explanation: DB2 encountered an error during the rebuild of the shared communications area (SCA) structure.

System action: DB2 stops.

System programmer response: DB2 uses the MVS IXLREBLD service during the SCA structure rebuild. Refer to *MVS/ESA Programming: Sysplex Services Reference* for return and reason codes from the IXLREBLD request.

DSN7512A *csect-name* **THERE IS A CRITICAL SHORTAGE OF FREE STORAGE IN SCA STRUCTURE** *sca-structure-name*

Explanation: The amount of free storage in the shared communications area (SCA) coupling facility structure is critically low. The SCA is 90 percent full.

This message stays on the MVS console and is highlighted.

System action: DB2 processing continues. However, DB2 limits use of SCA to critical information. Some utilities or commands that cannot be completed can cause one or more members of the group to come down.

This message might remain highlighted and might not be automatically deleted in a pre SP5.2.1-JES3 environment.

Operator response: If the message remains highlighted after the error situation is cleared up, manually delete the message from the console.

System programmer response: You can monitor the amount of SCA storage in use with the DISPLAY GROUP command.

You can free SCA storage by removing exception conditions. Another thing you can do is issue START DATABASE commands with the SPACENAM option or use the RECOVER utility to remove pages from the logical page list (LPL). Then restart any failed DB2s.

If your actions do not free up enough space, or if the problem continues to occur, you have the following options, depending on what level of MVS and the coupling facility control code you are running with:

- If all members of the group are running with MVS Version 5 Release 2 and with a coupling facility control code level greater than 0, issue the SETXCF START,ALTER command to increase the size of the SCA. The size you specify must be less than or equal to the maximum size defined the CFRM policy for the SCA structure.

  If the maximum size (SIZE in the CFRM policy) is still not big enough, take the actions described below for MVS Version 5 Release 1 systems.
- If any or all members of the group are running with MVS Version 5 Release 1 or with a coupling facility control code level of 0, increase the storage for the SCA in the CFRM policy. Then issue the MVS SETXCF START,REBUILD command to rebuild the structure.

Refer to *DB2 Data Sharing: Planning and Administration* for more detailed recovery information.

DSN7513I *csect-name* **SCA STRUCTURE** *sca-structure-name* **HAS ADEQUATE FREE STORAGE.**

Explanation: The shared communications area (SCA) coupling facility structure has adequate free storage to function properly.

This message is shown in the following circumstances:
- After the SCA was rebuilt.
- After the SCA usage goes below 67%, after its usage was up to 75% or higher.

System action: DB2 processing continues.

DSN7514I *csect-name* **RESPONSE TO AN MVS REBUILD RELATED EVENT FOR THE SCA STRUCTURE** *sca-structure-name* **FAILED. MVS IXLEERSP RETURN CODE =** *xxxxxxxx*, **MVS IXLEERSP REASON CODE =** *yyyyyyyy*.

Explanation: DB2 encountered an error during the rebuild of the shared communications area (SCA) structure.

System action: DB2 terminates.

System programmer response: DB2 uses the MVS IXLEERSP service to respond to SCA structure rebuild. Refer to *MVS/ESA Programming: Sysplex Services Reference* for return and reason codes from the IXLEERSP request.

Chapter 28. DSN8... messages

DSN8000I **REQUEST IS:** *request*

Explanation: The input statement is listed as received.

System action: The request is listed.

User response: See if the input request is what you want.

DSN8001I *csect* **EMPLOYEE NOT FOUND**

Explanation: The 'l' or 'list' request found no employee data that match your request.

System action: No employees are listed.

User response: If no employee data was expected to be found, continue.

Problem determination: If the error is unexpected, list the contents of the employee and department views VEMP and VDEPT to determine why the entry is missing.

DSN8002I *csect* **EMPLOYEE SUCCESSFULLY ADDED**

Explanation: Employee data was added as requested.

System action: The request was executed successfully.

DSN8003I *csect* **EMPLOYEE SUCCESSFULLY ERASED**

Explanation: Data for an employee was erased as requested.

System action: The request was executed successfully.

DSN8004I *csect* **EMPLOYEE SUCCESSFULLY UPDATED**

Explanation: Data for an employee was updated as requested.

System action: The request was executed successfully.

DSN8005E *csect* **EMPLOYEE EXISTS ALREADY, ADD NOT DONE**

Explanation: A request was made to add data for a new employee, but the employee number already exists.

System action: No employee data is added.

User response: Display the existing employee data. If this is the desired employee, update the employee entry. Otherwise, choose a new employee number.

Problem determination: List the contents of the employee view, VEMP, and the department view, VDEPT.

DSN8006E *csect* **EMPLOYEE DOES NOT EXIST, ERASE NOT DONE**

Explanation: The erase requested was not performed. The employee number requested was not present in the table.

System action: No employee data is erased.

User response: Determine if the employee data to be erased should exist. Verify that the correct search criteria and data were entered. If appropriate, correct the entries and continue processing.

Problem determination: List the contents of the employee view, VEMP, and the department view, VDEPT.

DSN8007E *csect* **EMPLOYEE DOES NOT EXIST, UPDATE NOT DONE**

Explanation: The update requested was not performed. The employee number requested was not present in the table.

System action: No employee data is updated.

User response: Verify that the correct search criteria and data were entered.

Problem determination: List the contents of the employee table (VPHONE, VEMPLP, or EMP) and see if the employee number to be updated is in the table. Correct the update request or the table.

DSN8008I *csect* **NO EMPLOYEE FOUND IN TABLE**

Explanation: No employees match the selection criteria.

System action: No employee data is listed.

User response: If no employee data was expected, continue processing.

Problem determination: List the contents of the VPHONE view and the underlying tables.

DSN8011I *csect* **DEPARTMENT NOT FOUND**

Explanation: No departments match the selection criteria.

System action: No departments are listed.

© Copyright IBM Corp. 1983, 2008

User response: If no departments are expected to be found, continue.

Problem determination: If the error is unexpected, list the contents of the department views VDEPT and VDEPMG1 and their underlying tables to determine why the entry is missing.

DSN8012I *csect* **DEPARTMENT SUCCESSFULLY ADDED**

Explanation: A department was added as requested.

System action: The request was executed successfully.

DSN8013I *csect* **DEPARTMENT SUCCESSFULLY ERASED**

Explanation: Data for a department was erased as requested.

System action: The request was executed successfully.

DSN8014I *csect* **DEPARTMENT SUCCESSFULLY UPDATED**

Explanation: Data for a department was updated as requested.

System action: The request was executed successfully.

DSN8015E *csect* **DEPARTMENT EXISTS ALREADY, ADD NOT DONE**

Explanation: You requested that a new department be added, but the department exists already.

System action: No departments are added.

User response: Display data for the existing department. If it is what you want, continue. If not, you can update data for this department or choose a new department id.

Problem determination: If the error is unexpected, list the contents of the department views VDEPT and VDEPMG1 and their underlying tables to see if the data differs from what is expected.

DSN8016E *csect* **DEPARTMENT DOES NOT EXIST, ERASE NOT DONE**

Explanation: The erase request found no departments which match your request.

System action: No department data is erased.

User response: See if the department data to be erased should exist. Did you enter the correct combination of search criteria and data? If not, correct the entries and continue.

Problem determination: If the error is unexpected, list the contents of the department views VDEPT and VDEPMG1 and their underlying tables to determine why the entry is missing.

DSN8017E *csect* **DEPARTMENT DOES NOT EXIST, UPDATE NOT DONE**

Explanation: The update request found no department data which matches your request.

System action: No department data is updated.

User response: See if the department data to be updated should exist. If you did not enter the correct combination of search criteria and data, correct the entries and continue.

Problem determination: If the error is unexpected, list the contents of the department views VDEPT and VDEPMG1 and their underlying tables to determine why the entry is missing.

DSN8018I *csect* **CURRENT DEPARTMENT NOT FOUND**

Explanation: A request was made by a 'current' department to reference a 'higher' department. The 'current' department was not found.

System action: No department data is processed.

User response: If the 'current' department was not expected to exist, continue processing.

Problem determination: List the contents of the department views VDEPMG1, VEMP, VASTRDE1, and VASTRDE2 and their underlying tables.

DSN8019E *csect* **NO HIGHER DEPARTMENT EXISTS**

Explanation: A request was made by a 'current' department to reference a 'higher' level of the department structure. A 'higher' level does not exist for the requested 'current' department.

System action: No department data is processed.

User response: If a 'higher' department was not expected to be found, continue processing.

Problem determination: List the contents of the department views VDEPMG1, VEMP, VASTRDE1, and VASTRDE2 and their underlying tables.

DSN8021I *csect* **ACTIVITY NOT FOUND**

Explanation: No activities match the selection criteria.

System action: No activity is listed.

User response: If no activity was expected to be found, continue processing.

Problem determination: List the contents of the activity views VPROJRE1, VACT, and VPROJACT and their underlying tables.

DSN8022I *csect* **ACTIVITY SUCCESSFULLY ADDED**

Explanation: An activity was added as requested.

System action: The request was executed successfully.

DSN8023I *csect* **ACTIVITY SUCCESSFULLY ERASED**

Explanation: An activity was erased as requested.

System action: The request was executed successfully.

DSN8024I *csect* **ACTIVITY SUCCESSFULLY UPDATED**

Explanation: An activity was updated as requested.

System action: The request was executed successfully.

DSN8025E *csect* **ACTIVITY EXISTS ALREADY, ADD NOT DONE**

Explanation: The addition requested was not performed. The activity requested was already present.

System action: No activity is added.

User response: If the activity was expected to already exist, continue processing.

Problem determination: List the contents of the activity views VPROJRE1, VACT, and VPROJACT and their underlying tables.

DSN8026E *csect* **ACTIVITY DOES NOT EXIST, ERASE NOT DONE**

Explanation: The erase requested was not performed. The activity requested was not found.

System action: No activity is erased.

User response: If the activity was not expected to be found, continue processing.

Problem determination: List the contents of the activity views VPROJRE1, VACT, and VPROJACT and their underlying tables.

DSN8027E *csect* **ACTIVITY DOES NOT EXIST, UPDATE NOT DONE**

Explanation: The update requested was not performed. The activity requested was not found.

System action: No activity is updated.

User response: Determine if the activity to be updated should exist. Verify that the correct search criteria and data were entered. If appropriate, correct the entries and continue processing.

Problem determination: List the contents of the activity views VPROJRE1, VACT, and VPROJACT and their underlying tables.

DSN8028I *csect* **ACTIVITY TYPE NOT FOUND, PLEASE ADD IT INTERACTIVELY**

Explanation: The add or update was not performed. The activity type entered was not found.

System action: No activity is listed.

User response: If the activity type was not expected to be found, add it interactively.

Problem determination: List the contents of the activity view, VACT, and its underlying tables.

DSN8031I *csect* **PROJECT NOT FOUND**

Explanation: No projects match the selection criteria.

System action: No projects are listed.

User response: If no projects are expected to be listed, continue.

Problem determination: If the error is unexpected, list the contents of the project view VPROJ and its underlying tables to determine why the entry is missing.

DSN8032I *csect* **PROJECT SUCCESSFULLY ADDED**

Explanation: A project was added as requested.

System action: The request was executed successfully.

DSN8033I *csect* **PROJECT SUCCESSFULLY ERASED**

Explanation: A project was erased as requested.

System action: The request was executed successfully.

DSN8034I *csect* **PROJECT SUCCESSFULLY UPDATED**

Explanation: The update was performed as requested.

System action: The request was executed successfully.

DSN8035E *csect* **PROJECT EXISTS ALREADY, ADD NOT DONE**

Explanation: You requested that a new project be added, but the project exists already.

System action: The request was not executed successfully.

User response: Display the existing project. If it is what you want, continue. If not, you can update this project or choose a new project id.

Problem determination: If the error was unexpected,

list the contents of the project view VPROJ and its underlying tables to see if the data differs from what is expected.

DSN8036E *csect* **PROJECT DOES NOT EXIST, ERASE NOT DONE**

Explanation: The erase request found no project which matches your request.

System action: No projects are erased.

User response: See if the project to be erased should exist. Did you enter the correct combination of search criteria and data? If not, correct the entries and continue.

Problem determination: If the error was unexpected, list the contents of the project view VPROJ and its underlying tables to determine why the entry is missing.

DSN8037E *csect* **PROJECT DOES NOT EXIST, UPDATE NOT DONE**

Explanation: The update request found no project which matches your request.

System action: No projects are updated.

User response: See if the project to be updated should exist. If you did not enter the correct combination of search criteria and data, correct the entries and continue.

Problem determination: If the error is unexpected, list the contents of the project view VPROJ and its underlying tables to determine why the entry is missing.

DSN8038I *csect* **CURRENT PROJECT NOT FOUND**

Explanation: A request was made by a 'current' project to reference a 'higher' project. The 'current' project was not found.

System action: No project data is processed.

User response: If the 'current' project was not expected to be found, continue processing.

Problem determination: List the contents of the project views VPROJRE1, VDEPT, VEMPPROJACT, VEMP, VSTAFAC1, VSTAFAC2, VPSTRDE1, and VPSTRDE2 and their underlying tables.

DSN8039E *csect* **NO HIGHER PROJECT EXISTS**

Explanation: A request was made by a 'current' project to reference a 'higher' level of the project structure. A 'higher' level does not exist for the requested 'current' project.

System action: No project data is processed.

User response: If a 'higher' project was not expected to be found, continue processing.

Problem determination: List the contents of the project views VPROJRE1, VDEPT, VEMPPROJACT, VEMP, VSTAFAC1, VSTAFAC2, VPSTRDE1, and VPSTRDE2 and their underlying tables.

DSN8041I *csect* **STAFFING NOT FOUND**

Explanation: No project staffing matches the selection criteria.

System action: No project staffing is listed.

User response: If no project staffing was expected to be found, continue processing.

Problem determination: List the contents of the project activity views VPROJRE1, VEMPPROJACT, VACT, and VEMP and their underlying tables.

DSN8042I *csect* **STAFFING SUCCESSFULLY ADDED**

Explanation: A project staffing was added as requested.

System action: The request was executed successfully.

DSN8043I *csect* **STAFFING SUCCESSFULLY ERASED**

Explanation: A project staffing was erased as requested.

System action: The request was executed successfully.

DSN8044I *csect* **STAFFING SUCCESSFULLY UPDATED**

Explanation: A project staffing was updated as requested.

System action: The request was executed successfully.

DSN8045E *csect* **STAFFING EXISTS ALREADY, ADD NOT DONE**

Explanation: A request was made to add a project staffing, but the project staffing already exists.

System action: No project staffing is added.

User response: Display the existing project staffing. If this is the desired project staffing, update it. Otherwise, choose a new project staffing.

Problem determination: List the contents of the project activity views VPROJRE1, VEMPPROJACT, VACT, and VEMP and their underlying tables.

DSN8046E *csect* **STAFFING DOES NOT EXIST, ERASE NOT DONE**

Explanation: The project staffing was not erased as requested.

System action: No project staffing is erased.

User response: Determine if the project staffing to be erased should exist. Verify that the correct search criteria and data were entered. If appropriate, correct the entries and continue processing.

Problem determination: List the contents of the project activity views VPROJRE1, VEMPPROJACT, VACT, and VEMP and their underlying tables.

DSN8047E *csect* **STAFFING DOES NOT EXIST, UPDATE NOT DONE**

Explanation: The update was not performed because the project staffing requested was not found.

System action: No project staffing is updated.

User response: Determine if the project staffing to be updated should exist. Verify that the correct search criteria and data were entered. If appropriate, correct the entries and continue processing.

Problem determination: List the contents of the project activity views VPROJRE1, VEMPPROJACT, VACT, and VEMP and their underlying tables.

DSN8051I *csect* **PROGRAM ENDED**

Explanation: The application program finished.

DSN8053I *csect* **ROLLBACK SUCCESSFUL, ALL UPDATES REMOVED**

Explanation: After any SQLWARNING or SQLERROR, all updates are removed via a ROLLBACK statement to prevent invalid updates.

System action: The ROLLBACK was executed successfully.

User response: Correct the prior error.

DSN8056I *csect* **NO MORE DATA TO DISPLAY**

Explanation: The last page of data has been displayed.

System action: No more data is shown on screen.

User response: If you expect no more data to be displayed, continue.

Problem determination: If more displayed data is expected, list the contents of the view and the underlying tables to see if the data differs from what is expected.

DSN8057I *csect* **FURTHER ENTRIES IN TABLE - UPDATE POSSIBLE**

Explanation: More data exists beyond that shown on the screen.

System action: No more data is shown on screen.

User response: Update entries as needed.

DSN8058I *csect* **PRESS PA1 FOR NEXT PAGE / ENTER FOR SELECTION MENU**

Explanation: Press the PA1 key to see the data on the next page or press the ENTER key to return to the selection menu.

User response: Press the PA1 key or the ENTER key depending on the function you want to perform.

DSN8060E *csect* **SQL ERROR, RETURN CODE IS: sqlcode**

Explanation: The SQL statement did not execute correctly. The 'sqlcode' can be found elsewhere in this manual. This may be an SQL error or an SQL warning.

System action: The request was not executed successfully.

User response: Refer to *DB2 Codes* for more information about the error.

Problem determination: Determine which SQL statement in the application program caused the error and see if it is reproducible. Follow the diagnosis approach suggested for the sqlcode.

DSN8061E *csect* **ROLLBACK FAILED, RETURN CODE IS: sqlcode**

Explanation: The SQL ROLLBACK statement was not executed correctly.

System action: The request was not executed successfully.

User response: Refer to *DB2 Codes*.

Problem determination: Determine which SQL statement in the application program caused the error and see if it is reproducible. Follow the diagnosis approach suggested for the sqlcode.

DSN8062E *csect* **AN OBJECT WAS NOT SELECTED**

Explanation: You did not specify an object code on the input line.

User response: Select one of the object codes listed on the panel.

DSN8064E *csect* **INVALID DL/I STC-CODE ON GU MSG:** *status-code*

Explanation: In attempting to get a message from IMS, an unexpected status-code was received.

System action: The application program is terminated.

Problem determination: Refer to the messages and codes manual for IMS/VS to determine the reason for the unexpected status code.

DSN8065E *csect* **INVALID DL/I STC-CODE ON ISRT MSG:** *status-code*

Explanation: In attempting to insert a message for IMS, an unexpected status-code was received.

System action: The application program is terminated.

Problem determination: Refer to the messages and codes manual for IMS/VS to determine the reason for the unexpected status code.

DSN8066E *csect* **UNSUPPORTED PFK OR LOGIC ERROR**

Explanation: The sample application found an unsupported program function key or logic error while trying to execute a request.

System action: The request was not executed successfully.

User response: Report the error in the application program.

DSN8067E *csect* **UNSUPPORTED SEARCH CRITERIA FOR OBJECT**

Explanation: You entered a search criteria value and an object value that do not find any selections.

System action: The request was not executed successfully.

User response: Enter new search criteria values.

Problem determination: If the values you entered are expected to provide valid selections, check the contents of the view and the underlying tables to see if the data differs from what is expected. If after checking the view and the underlying tables, you find the search criteria values you entered should provide valid selections, the error is in the sample program. Report the error.

DSN8068E *csect* **INVALID REQUEST SHOULD BE 'L' OR 'U'**

Explanation: The input must begin with 'L' or 'U'.

System action: The request is ignored.

User response: Correct the input request.

DSN8069E *csect* **NO VALID SELECTIONS QUALIFY FOR THIS REQUEST**

Explanation: You entered a search criteria value and a data values that do not find any selections.

System action: The request was not executed successfully.

User response: Enter new search criteria values.

Problem determination: If the criteria values you entered are expected to provide valid selections, check the contents of the view and the underlying tables to see if the data differs from what is expected. If after checking the view and the underlying tables, you find the search criteria values you entered should provide valid selections, the error is in the sample program. Report the error.

DSN8070E *csect* **VITAL DATA IS MISSING IN TABLE TOPTVAL**

Explanation: No valid entries exist in the table that contains possible valid entries.

System action: The request was not executed successfully.

User response: There is no valid entry you can enter. Report the error in the sample program.

DSN8072E *csect* **INVALID SELECTION ON SECONDARY SCREEN**

Explanation: You gave an invalid line number.

System action: The request was not executed successfully.

User response: You entered a number that is not in the list on the left under the heading NO for line number. Enter a valid number.

Problem determination: If the line number you entered is in the list on the left under the heading NO for line number, then the error is in the sample program.

DSN8073E *csect* **SPECIFIED LINE-NUMBER NOT FOUND IN PREVIOUS SCREEN**

Explanation: You gave an invalid line number.

System action: The request was not executed successfully.

User response: You entered a number that is not in the list on the left under the heading NO for line number. Enter a valid number. If the line number is in the previous screen, then this is an error in the sample program. Report the error.

DSN8074E *csect* **DATA IS TOO LONG FOR SEARCH CRITERIA**

Explanation: You entered a data value that has too many characters for the type of search criterion you selected. For example, department ids have a maximum of 3 characters and most other ids have a maximum of 6 characters.

System action: The request was not executed successfully.

User response: Enter a data value with the correct number of characters.

DSN8075E *csect* **MESSAGE FORMAT ROUTINE ERROR, RETURN CODE IS:** *sqlcode*

Explanation: There is an error in the parameters passed to the SQLCA formatting routine, DSNTIAR.

System action: The SQLCA could not be formatted properly.

User response: This is an error in the program. Check the return code listing of DSNTIAR in Part 2 of *DB2 Application Programming and SQL Guide*. Report the error.

DSN8076E *csect* **VITAL DATA IS MISSING IN TABLE 'TDSPTXT'**

Explanation: No valid entries exist in the table that contains possible valid entries.

System action: The request was not executed successfully.

User response: There is no valid entry you can enter. Report the error in the application program.

DSN8079E **CONNECTION TO DB2 NOT ESTABLISHED**

Explanation: Sample programs DSN8SC3, DSN8SP3, and DSN8HC3 cannot be called directly from TSO. Instead, they must be called from the ISPF sample connection manager, DSN8SCM or DSN8SPM.

System action: The program ends.

User response: Call the desired program through the connection manager, DSN8SCM or DSN8SPM.

DSN8080E **APPLICATION TERMINATED WITH CALL ATTACH REASON CODE** *reason-code*

Explanation: The call attachment facility service requested by the application program did not execute successfully.

System action: The requested service is not executed.

User response: Notify the system programmer.

System programmer response: Check the call attachment facility reason code.

Problem determination: Follow the failure analysis procedures for the call attachment facility reason code.

DSN8081E **RESULTS FROM THE CALL ATTACH TRANSLATE SERVICE**

Explanation: The call attachment facility OPEN service that was requested from the application program did not execute successfully. The TRANSLATE service was requested to provide information about the OPEN failure.

System action: The requested OPEN service was not executed.

User response: Refer to *DB2 Codes*.

Problem determination: Refer to *DB2 Codes* for more information about the error.

DSN8082E **THE DISCONNECT TERMINATED WITH REASON CODE** *reason-code*

Explanation: The call attachment facility DISCONNECT service requested by the application program did not execute successfully.

System action: The requested DISCONNECT service is not executed.

User response: Notify the system programmer.

System programmer response: Check the call attachment facility reason code.

Problem determination: Follow the failure analysis procedures for the call attachment facility reason code.

DSN8083E **AN UNKNOWN SERVICE** *service* **WAS REQUESTED**

Explanation: The user is requesting an invalid service.

System action: The requested service was not executed.

User response: Notify the system programmer.

System programmer response: The service request must be a valid call attachment facility service. Ensure that the request is one of the services supported by the call attachment facility.

Problem determination: There is an error in the Sample Application Program.

DSN8200E *csect* **INVALID DEPARTMENT NUMBER, EMPLOYEE NOT ADDED**

Explanation: The employee number that is to be added refers to a department number that is not contained in the table DEPT.

System action: The employee number is not added.

User response: Ensure that the correct department number was entered.

Problem determination: List the contents of the department table DEPT.

DSN8202E *csect* **EMPLOYEE NUMBER HAS DEPENDENT ROWS, NOT ERASED**

Explanation: The employee number to be erased is the same as a responsible employee number in the table PROJ or an employee number in the table EMPPROJACT. The employee number cannot be erased from the table EMP.

System action: No employee data is erased.

User response: Ensure that the correct employee number was entered.

Problem determination: List the contents of the PROJ and EMPPROJACT tables.

DSN8203E *csect* **INVALID WORK DEPT, EMPLOYEE NOT UPDATED**

Explanation: The employee that is to be updated has a department number that is not contained in the table DEPT. The update cannot be completed.

System action: No employee data is updated.

User response: Ensure that the correct department number was entered.

Problem determination: List the contents of the employee table DEPT.

DSN8210I *csect* **INVALID MGRNO, DEPARTMENT NOT ADDED**

Explanation: The department that is to be added has a manager number that is not contained in the table EMP. The add cannot be completed.

System action: No department is added.

User response: Ensure that the correct manager number was entered.

Problem determination: List the contents of the employee table EMP.

DSN8212E *csect* **DEPARTMENT NUMBER HAS DEPENDENT ROWS, NOT ERASED**

Explanation: The department number that is to be erased is the same as a project department number in table PROJ. The department cannot be erased.

System action: No department data is erased.

User response: Ensure that the correct department number was entered.

Problem determination: List the contents of the project table PROJ.

DSN8213E *csect* **INVALID ADMIN DEPT ID, DEPARTMENT NOT ADDED**

Explanation: The department that is to be added has a department number that is not contained in the table DEPT. The add cannot be completed.

System action: No department is added.

User response: Ensure that the correct department number was entered.

Problem determination: List the contents of the employee table DEPT.

DSN8214E *csect* **INVALID MANAGER ID, DEPARTMENT NOT UPDATED**

Explanation: The department that is to be updated has a manager number that is not contained in the table EMP. The update cannot be completed.

System action: No department data is updated.

User response: Ensure that the correct manager number was entered.

Problem determination: List the contents of the employee table EMP.

DSN8215E *csect* **INVALID ADMIN DEPT ID, DEPARTMENT NOT UPDATED**

Explanation: The department that is to be updated has a administrative department number that is not in the table DEPT. The update cannot be completed.

System action: No department data is updated.

User response: Ensure that the correct department number was entered.

Problem determination: List the contents of the employee table DEPT.

DSN8216E *csect* **DEPT NOT AT SPECIFIED LOCATION, EMPLOYEE NOT ADDED**

Explanation: The employee number that is to be added refers to a department number whose LOCATION field in the DEPT table is different from that specified for the employee. The add cannot be completed.

System action: No employee is added.

User response: Ensure that the correct department number was entered.

Problem determination: List the contents of the department table DEPT.

DSN8217E *csect* **DEPT NOT AT SPECIFIED LOCATION, EMP NOT UPDATED**

Explanation: The employee number that is to be updated refers to a department number whose LOCATION field in the DEPT table is different from that specified for the employee. The update cannot be completed.

System action: No employee data is updated.

User response: Ensure that the correct department number was entered.

Problem determination: List the contents of the department table DEPT.

DSN8220E *csect* **INVALID PROJECT NUMBER, ACTIVITY NOT ADDED**

Explanation: The activity that is to be added contains a project number that is not contained in the table PROJ. The add cannot be completed.

System action: The activity is not added.

User response: Ensure that the correct project number was entered.

Problem determination: List the contents of the project table PROJ.

DSN8222E *csect* **ACTIVITY HAS DEPENDENT ROWS, NOT ERASED**

Explanation: The activity that is to be erased from table PROJACT is the same as the corresponding activity in the table EMPPROJACT. The activity cannot be erased from the table PROJACT.

System action: No activity data is erased.

User response: Ensure that the correct activity was entered.

Problem determination: List the contents of the PROJACT and EMPPROJACT tables.

DSN8223E *csect* **ACTIVITY HAS DEPENDENT ROWS, NOT UPDATED**

Explanation: A start date in table PROJACT is to be updated, but there is a project staffing for this activity in table EMPPROJACT. The start date in table PROJACT cannot be updated.

System action: No activity data is updated.

User response: Ensure that the correct start date was entered.

Problem determination: List the contents of the project-activity-staffing table EMPPROJACT and project-activity table PROJACT.

DSN8230E *csect* **INVALID DEPARTMENT NUMBER, PROJECT NOT ADDED**

Explanation: The project that is to be added contains a department number that is not present in the table DEPT. The add cannot be completed.

System action: No project is added.

User response: Ensure that the correct department number was entered.

Problem determination: List the contents of the department table DEPT.

DSN8231E *csect* **INVALID RESPONSIBLE ID, PROJECT NOT ADDED**

Explanation: The project that is to be added contains an employee number that is not contained in the table EMP. The add cannot be completed.

System action: No project is added.

User response: Ensure that the correct employee number was entered.

Problem determination: List the contents of the employee table EMP.

DSN8233E *csect* **PROJECT NUMBER HAS DEPENDENT ROWS, NOT ERASED**

Explanation: A project number in table PROJ is to be erased, but it is the same as a project number in table PROJACT. The project cannot be erased from table PROJ.

System action: No project is erased.

User response: Ensure that correct project number was entered.

Problem determination: List the contents of the PROJ and PROJACT tables.

DSN8235E *csect* **INVALID MAJOR PROJECT ID, PROJECT NOT ADDED**

Explanation: The project that is to be added contains a major project ID that is not contained in the table PROJ. The add cannot be completed.

System action: No project is added.

User response: Ensure that the correct major project number was entered.

Problem determination: List the contents of the project table PROJ.

DSN8236E *csect* **INVALID ASSOCIATE DEPT ID, PROJECT NOT UPDATED**

Explanation: The project that is to be updated has a department number that is not contained in the table DEPT. The update cannot be completed.

System action: No project data is updated.

User response: Ensure that the correct department number was entered.

Problem determination: List the contents of the department table DEPT.

DSN8237E *csect* **INVALID RESPONSIBLE ID, PROJECT NOT UPDATED**

Explanation: The project that is to be updated has a employee number that is not contained in the table EMP. The update cannot be completed.

System action: No project data is updated.

User response: Ensure that the correct employee number was entered.

Problem determination: List the contents of the employee table EMP.

DSN8238E *csect* **INVALID MAJOR PROJECT ID, PROJECT NOT UPDATED**

Explanation: The project that is to be updated has a major project number that is not contained in the table PROJ. The update cannot be completed.

System action: No project data is updated.

User response: Ensure that the correct project number was entered.

Problem determination: List the contents of the project table PROJ.

DSN8240E *csect* **INVALID EMPLOYEE NUMBER, STAFFING NOT ADDED**

Explanation: The staffing that is to be added refers to an employee number that is not present in table EMP. The add cannot be completed.

System action: No staffing is added.

User response: Ensure that the correct employee number was entered.

Problem determination: List the contents of employee table EMP.

DSN8242E *csect* **INVALID PROJECT ACTIVITY DATA, STAFFING NOT ADDED**

Explanation: The staffing that is to be added refers to a project number, activity number, and start date that is not present in table PROJACT. The add cannot be completed.

System action: No staffing is added.

User response: Ensure that the correct project number, activity number, and start date were entered.

Problem determination: List the contents of the project table PROJACT.

DSN8243E *csect* **INVALID START DATE, STAFFING NOT UPDATED**

Explanation: The start date given is not currently in the project-activity table PROJACT. The update cannot be completed.

System action: No staffing data is updated.

User response: Ensure that the correct start date was entered.

Problem determination: List the contents of the project-activity table PROJACT and the project-activity-staffing table EMPPROJACT.

Chapter 29. DSN9... messages

DSN9000I KEYWORD *keyword-name* APPEARS MORE THAN ONCE

Explanation: The same keyword (specified in the messages as 'keyword-name') appears more than once in the command. This message will be issued for each occurrence of the keyword after the first.

System action: Processing for the command is terminated.

System programmer response: Correct the command by using the keyword only once, and reenter the command.

Problem determination: Obtain a copy of the system or terminal log to verify the command entry.

DSN9001I KEYWORD *keyword-name* IS INVALID

Explanation: The keyword (specified in the message as 'keyword-name') is unknown or undefined. The keyword may be misspelled, or the keyword may not be applicable to the command being processed.

System action: Processing for the command is terminated.

System programmer response: Correct the keyword, and reissue the command.

Problem determination: Obtain a copy of the system or terminal log to verify the command entry.

DSN9002I UNBALANCED PARENTHESES WERE FOUND FOLLOWING KEYWORD *keyword-name*

Explanation: An invalid combination of parentheses has been encountered following the keyword (specified in the message as 'keyword-name'). Parenthesis must be paired open and close, and a close parenthesis must follow an open parenthesis before another open parenthesis is encountered.

System action: Processing for the command is terminated.

System programmer response: Correct the error, and reissue the command.

Problem determination: Obtain a copy of the system or terminal log to verify the command entry.

DSN9003I KEYWORD *keyword-name* PARAMETER CONTAINS UNBALANCED APOSTROPHES

Explanation: An odd number of apostrophes is present in a parameter. If the parameter is a quoted string, it must have one apostrophe at each end of the string. If an apostrophe is to appear within the string, two adjacent apostrophes must be entered. If the parameter is a hexadecimal value, it must be entered as X'hex characters'.

System action: Processing for the command is terminated.

System programmer response: Correct the error, and reissue the command.

Problem determination: Obtain a copy of the system or terminal log to verify the command entry.

DSN9004I KEYWORD *keyword-name* RANGE (c) INCORRECTLY SPECIFIED

Explanation: The range for a keyword (specified in the message as 'keyword-name') was incorrectly specified using the syntax character specified in the message as 'c'. The standard syntax character used to denote a range is a colon (:). Either the ending limit of the range was omitted or a subset was specified within the range.

Examples of this error are LINE(1:) and GROUP(A*:B*).

Examples of a correctly specified range are LINE(1:3) and GROUP(A:B).

System action: Processing for the command is terminated.

System programmer response: Refer to *DB2 Command Reference* to verify that the command you are using allows a range for the given keyword. Correct the error, and reissue the command.

Problem determination: Obtain a copy of the system or terminal log to verify the command entry.

DSN9005I KEYWORD *keyword-name* SUBSET (c) INCORRECTLY SPECIFIED

Explanation: The subset for a keyword (specified in the message as 'keyword-name') was incorrectly specified for use with the syntax character 'c' in the message. The standard syntax character used to denote a subset is an asterisk (*).

For example, consider a group of objects named BLACK, BLUE, GREEN, RED, and WHITE. You may either explicitly specify BLACK and BLUE, or you may specify BL*, meaning all members of the group whose names begin with BL. The character that follows the asterisk must be a blank, comma, close parenthesis, or colon.

© Copyright IBM Corp. 1983, 2008

Examples of such an error are NAME(BL*CK) and NAME(*LUE).

Examples of correct specification are NAME(BL*) and UNIT(2*).

System action: Processing for the command is terminated.

System programmer response: Refer to *DB2 Command Reference* to be sure that the command you are using allows a subset for the named keyword. Correct the error, and reissue the command.

Problem determination: Obtain a copy of the system or terminal log to verify command entry.

DSN9006I KEYWORD *keyword-name* **ASTERISK (c) INCORRECTLY USED**

Explanation: For the keyword specified in the message (as 'keyword-name'), an asterisk (c) was used as the first character of a parameter to specify ALL. However, the asterisk appears in a list, or the characters in juxtaposition are not a blank, comma, equal sign, or parentheses.

Examples of such errors are LINE(LINE1,*,LINE7), NAME(**), and GROUP(PQR:*).

Examples of correct specification are UNIT(*) and LINE=* etc.

System action: Processing for the command is terminated.

System programmer response: Refer to *DB2 Command Reference* to be sure the command you are using allows specification of ALL for the given keyword. Correct the error, and reissue the command.

Problem determination: Obtain a copy of the system or terminal log to verify command entry.

DSN9007I THIS COMMAND REQUIRES ONE OF THE FOLLOWING KEYWORDS BE SPECIFIED, *keyword1 keyword2 keyword3 keyword4 keyword5 keyword6 keyword7 keyword8*

Explanation: One of the keywords identified in the list must be specified on this command, but none of these keywords was present.

System action: Processing for the command is terminated.

System programmer response: Identify the appropriate keyword from the list, include that keyword, and reissue the command. Refer to *DB2 Command Reference* for descriptions of the keywords.

Problem determination: Obtain a copy of the system or terminal log to verify command entry.

DSN9008I KEYWORD *keyword-name* **MAY NOT BE NEGATED**

Explanation: The negation characters (NO) appear in front of the keyword (specified in the message as 'keyword-name'), but negating this keyword is not allowed. As an example, consider a keyword called PRINT that could indicate that you want something printed. If you did not want the material to be printed, you might specify NOPRINT. This process of prefixing a keyword with the characters NO is called negation. While negation may be meaningful on some keywords, it is not allowed on all keywords.

System action: Processing for the command is terminated.

System programmer response: Reenter the command, but do not negate the named keyword. Refer to *DB2 Command Reference* for further information concerning this command.

Problem determination: Obtain a copy of the system or terminal log to verify the command entry.

DSN9009I KEYWORD *keyword-name* **IS MISSING AND MUST BE SPECIFIED**

Explanation: The keyword specified in the message (as 'keyword-name') must be present, but it was not entered. This keyword must be present in order for the command to process properly.

System action: Processing for the command is terminated.

System programmer response: Enter the command again, and include the specified keyword.

Problem determination: Obtain a copy of the system or terminal log to verify the command entry.

DSN9010I REQUIRED PARAMETER FOR KEYWORD *keyword-name* **IS MISSING**

Explanation: A parameter must be specified for the keyword specified in the message (as 'keyword-name'), but no parameter was entered.

As an example of this error, consider a keyword called PRIORITY whose value can range from 0 to 9. Specification of PRIORITY(1) is acceptable, but, if you specify PRIORITY with no parameter value, the keyword is meaningless.

System action: Processing for the command is terminated.

System programmer response: Supply appropriate parameters for the specified keyword and reissue the command. Refer to *DB2 Command Reference* for valid parameter values.

Problem determination: Obtain a copy of the system or terminal log to verify the command entry.

DSN9011I PARAMETER(S) CANNOT BE SPECIFIED FOR KEYWORD *keyword-name*

Explanation: No parameter(s) can be specified for the keyword specified in the message (as 'keyword-name'). You may have misspelled the keyword-name, causing it to be interpreted as a different keyword, or the parameter may actually belong with another keyword instead of the one on which you entered it.

System action: Processing for the command is terminated.

System programmer response: Correct the error, and reissue the command. Refer to *DB2 Command Reference* for details on how to enter the command.

Problem determination: Obtain a copy of the system or terminal log to verify the command entry.

DSN9012I KEYWORD *keyword-name* **PARAMETER(S) MUST BE HEXADECIMAL**

Explanation: Parameter values for the keyword specified in the message (as 'keyword-name') must be hexadecimal values. The value must be specified as X'hex characters'. Hexadecimal characters are defined as the numeric digits 0 through 9 and the letters A through F. Either uppercase or lowercase is acceptable for the letters A through F.

As an example of this error, consider the keyword OFFSET, for which you are expected to enter a hexadecimal value. If you enter OFFSET(X'2C'), it is acceptable. However, entering OFFSET(Jones) is not acceptable.

System action: Processing for the command is terminated.

System programmer response: Reissue the command, but ensure that the parameters for the named keyword are hexadecimal values.

Problem determination: Obtain a copy of the system or terminal log to verify the command entry.

DSN9013I KEYWORD *keyword-name* **PARAMETER** *parameter value* **EXCEEDS ALLOWABLE LENGTH OF** *nn*

Explanation: The parameter value (specified in the message as 'parameter value') exceeds the limit of 'nn' characters in length.

As an example of this type of error, consider a PRIORITY keyword whose value may range from zero to nine. If you specify PRIORITY(21) or PRIORITY(006), the value you have entered exceeds the limit of one character.

System action: Processing for the command is terminated.

System programmer response: Refer to *DB2 Command Reference* for a list of acceptable parameters. Correct the error, and reissue the command.

Problem determination: Obtain a copy of the system or terminal log to verify the command entry.

DSN9014I KEYWORD *keyword-name* **LIMIT OF** *nn* **PARAMETER(S) EXCEEDED**

Explanation: Too many parameters have been specified for the keyword that is specified in the message. The command processor does not accept more than the limit (specified in the message as 'nn') of parameters. This error is probably caused by entering too many parameters. It could also be caused by a missing close parenthesis that has not yet been detected.

System action: Processing for the command is terminated.

System programmer response: Reissue the command, and enter no more than the specified limit of parameters for the given keyword. If necessary, issue the command more than once to process all the desired parameter values. Refer to *DB2 Command Reference* for further details.

Problem determination: Obtain a copy of the system or terminal log to verify the command entry.

DSN9015I PARAMETER *parameter-value* **IS UNACCEPTABLE FOR KEYWORD** *keyword-name*

Explanation: The parameter value specified in the message is not an acceptable value for the named keyword.

As an example, consider a keyword called PRINT that can have parameters of YES or NO. Specification of PRINT(YES) or PRINT(NO) is valid. However, specification of PRINT(MAYBE) is unacceptable.

System action: Processing for the command is terminated.

System programmer response: Refer to *DB2 Command Reference* for a list of acceptable parameters. Correct and reissue the command.

Problem determination: Obtain a copy of the system or terminal log to verify the command entry.

DSN9016I *verb-name* **COMMAND REJECTED, UNAUTHORIZED REQUEST**

Explanation: The command specified in the message (as 'verb-name') requires that you have proper authorization in order to use the command. You do not have the required authorization.

System action: The command is not executed. Processing is terminated.

Operator response: If the command must be executed on behalf of the user and your installation operating procedures permit it, enter the command on request.

System programmer response: Contact the system programmer responsible for system security, and request that this person grant you authorization to use the command. Otherwise, you must have someone who is authorized issue the command for you. If necessary, request the system operator to enter the command for you.

Problem determination: Obtain a copy of the system or terminal log to verify the command entry.

DSN9017I FAILURE WHILE PROCESSING *verb-name pkw-name* **COMMAND, PROCESSING TERMINATED**

Explanation: The DB2 command processor has abended while processing the command (specified in the message as 'verb-name pkw-name'). The error is been recorded in SYS1.LOGREC, and an SVC dump is requested. The command may have partially completed. Refer to any prior response messages to determine what has been done.

System action: Processing for the command is terminated.

Operator response: If a dump was produced, have it printed. Print a copy of the contents of the SYS1.LOGREC data set for the time period involved. Forward these items with a copy of the system log to the system programmer.

System programmer response: Reissue the command. If a second failure occurs, consult your supervisor.

Problem determination: Obtain a copy of the system or terminal log to verify command entry. Contact the system operator to make sure that diagnostic information is forwarded to you.

DSN9018I *csect-name* **ABORTING** *verb-name* **PROCESSING DUE TO INSUFFICIENT STORAGE AVAILABLE**

Explanation: The command processor was unable to obtain sufficient storage to complete processing of any response messages generated by the invoked command.

System action: Processing for the command is terminated abnormally.

Operator response: Notify the system programmer before attempting to reissue the command.

System programmer response: Determine the reason for insufficient storage available to the command. If a cursory examination of the system indicates that the region sizes for DB2 are sufficient, and the current workload does not seem excessive, then a MVS Console dump (MVS DUMP command) may be required. Insure that any retry steps defined by the command are performed prior to reissuing the command.

Problem determination: The invoked command had completed processing and returned to the Command Preprocessor when an attempt was made to obtain storage from the address space from which the command was entered. Since sufficient storage was unavailable, no response messages from the invoked command are available.

DSN9019I VERB *verb-name* **IS NOT A KNOWN COMMAND**

Explanation: The command represented by the verb specified in the message is not known to the system; it is an undefined command. A possible cause for this error is that you made a spelling error while entering the command.

System action: Processing for the command is terminated.

System programmer response: Refer to *DB2 Command Reference* for the correct command format.

Problem determination: Obtain a copy of the system or terminal log to verify the command entry.

DSN9020I ONLY ONE OF THE FOLLOWING KEYWORDS MAY BE SPECIFIED ON THIS COMMAND, *keyword1 keyword2 keyword3 keyword4 keyword5 keyword6 keyword7 keyword8*

Explanation: Only one of the identified keywords can be specified at the same time on this command.

System action: Processing for the command is terminated.

System programmer response: Identify the appropriate keyword from the list, include only that keyword, and reissue the command. Refer to *DB2 Command Reference* for descriptions of the keywords.

Problem determination: Obtain a copy of the system or terminal log to verify the command entry.

DSN9021I VERB *verb-name* **REQUIRED KEYWORD IS MISSING**

Explanation: The command specified in the message (as 'verb-name') was entered, but at least one required keyword is missing. The command processor was unable to determine which keyword or keywords were missing.

System action: Processing for the command is terminated.

System programmer response: Refer to *DB2 Command Reference* for the proper format of the named command, and then reenter the command.

Problem determination: Obtain a copy of the system or terminal log to verify command entry. If a failure is suspected, take a storage dump.

DSN9022I *csect-name verb-name pkw-name* **NORMAL COMPLETION**

Explanation: All synchronous processing for the command specified in the message as 'verb-name pkw-name' completed successfully. Any tasks executing asynchronously for the specified command might still be executing when this message is displayed.

System action: Synchronous processing for the specified command is complete. Subsequent system actions defined by the command are documented in *DB2 Command Reference* .

DSN9023I *csect-name verb-name pkw-name* **ABNORMAL COMPLETION**

Explanation: The command specified in the message as 'verb-name pkw-name' completed unsuccessfully. The command issued one or more error messages before this message. The name of the CSECT that issued the message is shown at the beginning of the message as a diagnostic aid.

System action: Processing for the command ended.

System programmer response: Proceed as indicated by previous error messages.

DSN9024I **KEYWORD** *keyword-name* **PARAMETER MAY NOT SPECIFY A RANGE OF VALUES**

Explanation: The parameter of the keyword specified in the message specifies a range using the colon operator, (for instance, 1:3), but a range of values is not allowed.

As an example, consider a keyword called CLASS. You might enter CLASS(1,2,3) to specify classes 1, 2, and 3, or you might enter CLASS(1:3). However, not all commands allow you to specify a range of values.

System action: Processing for the command is terminated.

System programmer response: Correct the keyword parameter, and reenter the command.

Problem determination: Obtain a copy of the system or terminal log to verify command entry.

DSN9025I *csect-name verb-name pkw-name* **NORMAL COMPLETION FOR SYNCHRONOUS PHASE, AND IN PROGRESS FOR ASYNCHRONOUS PHASE.**

Explanation: All synchronous processing for the command specified as 'verb-name pkw-name' in this message completed successfully. Any tasks executing asynchronously for the specified command might still be executing when this message is displayed.

System action: Synchronous processing for the specified command is complete. Subsequent system actions defined by the command that can occur during the asynchronous phase of command processing are documented in *DB2 Command Reference*.

DSN9026I **KEYWORD** *keyword-name* **PARAMETER(S) MUST BE ALPHABETIC**

Explanation: Parameter values for the KEYWORD *keyword-name* must be all alphabetic characters. *keyword-name* can consist of any combination of the 26 uppercase (A through Z) and 26 lowercase (a through z) letters of the English alphabet plus the three characters reserved as alphabetic extenders for national languages ($, #, and @).

An example of this type of error, is if in the keyword NAME field, you are expected to enter a person's last name. If you enter NAME(Jones) it is acceptable, but if you enter NAME(Jon4s), it is not acceptable because it contains numeric characters.

System action: Processing for the command is terminated.

System programmer response: Reissue the command but ensure that the parameters for the named keyword are of the required type.

Problem determination: Obtain a copy of the system or terminal log to verify the command entry.

DSN9027I **KEYWORD** *keyword-name* **PARAMETER(S) MUST BE ALPHAMERIC**

Explanation: Parameter values for the keyword *keyword-name* must consist of all numbers and alphabetic characters. *keyword-name* can consist of any combination of the 26 uppercase (A through Z) and 26 lowercase (a through z) letters of the English alphabet, plus the three characters reserved as alphabetic extenders for national languages ($, #, and @), as well as numeric characters.

System action: Processing for the command is terminated.

System programmer response: Reissue the command but ensure that the parameters for the named keyword are of the required type.

Problem determination: Obtain a copy of the system or terminal log to verify the command entry.

DSN9028I KEYWORD *keyword-name* **PARAMETER(S) MUST BE NUMERIC**

Explanation: Parameter values for the keyword (specified on these messages) must be of a specific type, such as all ALPHABETIC or all ALPHAMERIC / NATIONAL characters, or all NUMERIC values.

As an example of this type of error, consider the keyword NAME, where you are expected to enter a person's last name. If you enter NAME(Jones) it is acceptable, but if you enter NAME(Jon4s), it is not acceptable.

As a second example, consider a keyword WEIGHT that specifies a numeric weight. Entering WEIGHT(47) is acceptable, but entering WEIGHT(4u) is not.

System action: Processing for the command is terminated.

System programmer response: Reissue the command but ensure that the parameters for the named keyword are of the required type.

Problem determination: Obtain a copy of the system or terminal log to verify the command entry.

DSN9029I *csect-name* **FAILURE WHILE PROCESSING A COMMAND**

Explanation: A failure has been encountered while processing a command. The command may or may not have been executed. The error has been recorded in the system error log (the SYS1.LOGREC data set), and an SVC dump was attempted.

System action: Processing for the command is terminated.

Operator response: If a dump was produced, have it printed. Print a copy of the contents of the SYS1.LOGREC data set for the time period involved. Forward these items with a copy of the system log to the system programmer.

System programmer response: Reissue the command. If a second failure occurs, consult your supervisor or the system programmer.

Problem determination: Obtain a copy of the system or terminal log to verify command entry. Contact the system operator to make sure that diagnostic information is forwarded to the system programmer.

DSN9030I KEYWORD *keyword-name* **PARAMETER MAY NOT SPECIFY A SUBSET OF VALUES**

Explanation: The parameter for the keyword specified in the message identifies a subset using the asterisk operator (for instance, ABC*, *ABC, or *ABC*DEF*), but using a subset is not allowed for this command.

As an example, consider a keyword called NAME, for which you can specify people's names. You might specify NAME(ROBERT ROBERTS ROBERTSON) to enter three names, or you might specify NAME(ROB*) to indicate all names that begin with the characters ROB. NAME(*BERT*) could also be specified to indicate all names that contain the characters BERT within the name. However, not all commands allow you to specify a subset of values in this way.

System action: Processing for the command is terminated.

System programmer response: Correct the keyword parameter, and reenter the command.

Problem determination: Obtain a copy of the system or terminal log to verify command entry.

DSN9031I SYNTAX ERROR DETECTED FOLLOWING KEYWORD *keyword-name*

Explanation: The text that follows the named keyword contains invalid syntax.

As an example, consider specification of CLASS==A. The double equal signs do not conform to the rules of syntax and make the keyword meaningless. This is just one example of invalid syntax. It is not practical to try to illustrate every possible error here. You could, however, correctly specify CLASS=A or CLASS(A).

System action: Processing for the command is terminated.

System programmer response: Examine the text following the named keyword to ensure that no errors were made as the command was entered. Ensure that you have followed the rules for command entry, and reenter the command.

Problem determination: Obtain a copy of the system or terminal log to verify command entry.

DSN9032I REQUESTED FUNCTION IS NOT AVAILABLE

Explanation: An attempt was made to invoke a command processor that was not loaded. An example of this is invoking -START DDF or -STOP DDF when DDF is not loaded (not selected during the installation process).

System action: The requested function is not performed.

User response: Determine what function was requested. Do not invoke this function until DB2 is reinstalled and DDF is selected.

Problem determination: Look at the console log to determine which command caused the error.

DSN9033I KEYWORD *name1* REQUIRES KEYWORD *name2*

Explanation: When keyword 'name1' is specified, keyword 'name2' must also be specified.

System action: Processing for the command terminates.

User response: Resubmit the command with the missing keyword.

Problem determination: Obtain a copy of the system or terminal log to verify the command entry.

DSN9034I THE TOTAL LENGTH OF COMMAND RESPONSE MESSAGES EXCEEDED ITS LIMIT. DISPLAY IS TERMINATED.

Explanation: The total length of all response messages returned from a single DB2 command exceeded the internal limit set by the system. The message chain was truncated. This message was appended at the end of the chain to inform the user of the termination of the report.

DSN9035I BEGIN OF DISPLAY FOR MEMBER:
memname

Explanation: This message precedes the command
output from another member of a data sharing group
as the result of the SCOPE(GROUP) keyword. The
message number is not included in this message.

memname
Member name for which command output is
being displayed.

System action: Processing continues normally.

DSN9036I

Explanation: This message underscores message
DSN9035I and separates the command output from
another member of a data sharing group as the result
of the SCOPE(GROUP) keyword. The message number
is not included in this message.

System action: Processing continues normally.

DSN9037I END OF DISPLAY FOR MEMBER:
memname

Explanation: This message follows messages
DSN9035I, DSN9036I, and the command output from
another member of a data sharing group as the result
of the SCOPE(GROUP) keyword and indicates that the
output was generated successfully. The message
number is not included in this message.

memname
Member name for which command output is
being displayed.

System action: Processing continues normally.

DSN9038I *csect-name* COMMAND REJECTED BECAUSE DB2 WAS RESTARTED IN LIGHT MODE

Explanation: This command is not allowed when the DB2 member has been restarted with the LIGHT(YES) parameter.

System action: Command processing is halted.

System programmer response: Refrain from issuing this command on this DB2 member until after the member has been restarted without LIGHT(YES). If the command has group scope, then you may be able to issue the command to a different DB2 member in the group.

Part 3. Appendixes

© Copyright IBM Corp. 1983, 2008

Appendix A. Problem determination

This section contains the following topics about problem determination:

- "Diagnostic items"
- "Resource types" on page 662
- "ABEND codes associated with DSN1COPY misuse" on page 665
- "Explanations" on page 667

Diagnostic items

1. Console output from the system on which the job was run, and a listing of the SYSLOG data set for the period of time spanning the failure.
2. Dynamic dump, taken to SYS1.DUMPxx data set, by DB2 (04E/04F abends).
3. SVC dump (or system dump), taken to SYS1.DUMPxx data set, as result of an operator initiated dump command or SLIP trap exit.
4. SYSABEND, SYSUDUMP, or SYSMDUMP output.
5. Listing of SYS1.LOGREC data set, obtained by executing IFCEREP1.
6. Listing of DB2 log output.
7. Link pack area map (LPAMAP) and nucleus load module map, obtained by executing AMBLIST.
8. Listing of the procs used to initialize DB2 address spaces.
9. Listing of DSNZPARM module used.
10. IDCAMS LISTCAT output for the ICF (VSAM) catalog containing the DB2 subsystem catalog that defined the failing index or table.
11. IDCAMS LISTCAT output showing the attributes of all of the active log data sets for the member of the DB2 member that failed.
12. VTOC listing of the DASD volume on which the failing operation occurred.
13. Dump of the index or table space that failed.
14. Listing of the results produced by the SQL statements.
15. Listing of DB2 catalog tables.
16. Contents of the BSDS (both copies if dual BSDS in use) obtained by using the DB2 print log map (DSNJU004) utility.
17. Source listing of the failing application program.
18. IRLM trace output.
19. Output of the GTF trace (if requested by IBM).
20. Listing of the DSNTRACE data set for the DSN (TSO attachment facility) session involved in the problem.
21. Listing of the ISPF log for the session involved in the problem.
22. CICS trace listing.
23. CICS attach MVS SNAP dump.
24. CICS transaction dump.
25. IMS trace output.
26. IMS dependent region and/or control region dump.
27. IMS/VS Version 1 system log information, for the period of time spanning the failure.
28. VVDS dump of the volumes containing the data set.
29. DB2 Repair dump of the requested page.
30. DSN1LOGP detail report containing log records associated with the damaged page.
31. Detailed description of events immediately prior to and during the abend.

© Copyright IBM Corp. 1983, 2008

32. SYSPRINT output, including JCL, for the application program or batch job and system messages that were issued. Make sure that MSGLEVEL=(1,1) on the JOB statement so that all diagnostic information is sent to SYSPRINT.
33. DB2 log output, including any DB2 Recovery Log Archive Tapes.
34. DB2 dump of the requested page.
35. Definitions of views and tables involved in the error.
36. Listing of the IRLM procedure.
37. Definitions of table spaces, tables, and indexes involved in the error.
38. Results from SELECT * from SYSIBM.SYSDATABASE.
39. Use IDCAMS to print from BSDS the record with key of X'16000001'.
40. DB2 Repair dump of the pages identified in DSNI014I messages.
41. CHECK utility output.
42. DB2 Repair dump of the pages identified in DSNI013I messages.
43. A copy via (DSN1COPY) of the index space or table space identified in the DSNI013I messages on tape.
44. DB2 Repair dump of the header page (page 0) for each index identified in the DSNI013I messages.
45. IMS console log.
46. UTPRINT output produced by utility invocation of SORT.
47. Repair dump of all SYSUTIL pages.
48. Dump of page set from DUMP option of REPAIR.
49. All dumps taken by the attachment facility or requesting application.
50. Console output from the system on which the job was run, and a listing of the MVS SYSLOG data set for the period from the last IPL to the time of the failure.
51. An MVS stand-alone dump.
52. Precompiler listing of failing application program with SOURCE(YES) and XREF(YES) specified.
53. Listing of SYSIBM.SYSLGRNX and/or SYSLGRNG, and SYSIBM.SYSCOPY.
54. A copy of the DDITV02 SYSIN data set.
55. A copy of the DDOTV02 output data set.
56. The console output and the listing of the SYSLOG data set from each connected site for the period of time spanning the failure.
57. The listings of the SYS1.LOGREC data set, obtained by executing IFCEREP1 at each connected site.
58. A listing of the contents of the SYSIBM.IPNAMES and SYSIBM.LUNAMES tables at the local DB2 system and the remote DB2 system.
59. A listing of the contents of the SYSIBM.USERNAMES table at the local DB2 system and the remote DB2 system.
60. Dynamic dump, taken to SYS1.DUMPxx data set, by DB2 (04E/04F abends). Availability of this dump is dependent upon the failing environment.
61. SYSABEND, SYSUDUMP, or SYSMDUMP output. Availability of these dumps is dependent upon the failing environment.
62. A copy (via DSN1COPY) on tape of the page set containing the DB2 Directory. Alternatively, request a dump (produced by the REPAIR Utility, DSN1COPY or DSN1PRNT) of the DBD for each database identified in the DSNI013I messages.
63. Listing of SYS1.LOGREC data set, obtained by executing IFCEREP1 and ISDASDA0.
64. A copy of the table and index taken via DSN1COPY before any recover is performed.
65. Detailed description of events prior to the abend such as recovery or other activity affecting DSNDB06.SYSDBASE, DSNDB01.DSD01, or the subject table spaces.

66. Definitions of table spaces, tables, and indexes involved in the error. The record identifiers (OBIDs) identified with each table in the table space with the error can be obtained from SYSIBM.SYSTABLES.
67. Listing of the entire table space.
68. Listing of the DBD, obtained by executing UTILITY DIAGNOSE OBD.
69. Unformatted DSN1PRNT output of the DSNDB01.DBD01 page set. Use the method described in Part 5 of *DB2 Diagnosis Guide and Reference* to locate the inconsistent DBD in the DSNDB01.DBD01 page set.
70. The results of the following SQL:
    - SELECT * FROM SYSIBM.SYSINDEXES WHERE DBID=dbid ORDER BY OBID; Where dbid is the DBID of the damaged DBD.
    - SELECT * FROM SYSIBM.SYSINDEXES WHERE DBID=dbid ORDER BY ISOBID;
    - SELECT * FROM SYSIBM.SYSLINKS WHERE DBID=dbid ORDER BY OBID;
    - SELECT * FROM SYSIBM.SYSTABLES WHERE DBID=dbid ORDER BY OBID;
    - SELECT * FROM SYSIBM.SYSTABLESPACE WHERE DBID=dbid ORDER BY OBID;
    - SELECT * FROM SYSIBM.SYSTABLESPACE WHERE DBID=dbid ORDER BY PSID.

    Use the results to compare the information in the DB2 Catalog with the DBD. Determine which OBDs are inconsistent with the DB2 Catalog. Use Part 5 of *DB2 Diagnosis Guide and Reference* to analyze the DBD.
71. DSN1LOGP output with DBID (1) OBID (1F) covering the period of time in which the DBD inconsistency may have occurred. This period of time is roughly from the time of the ABEND back to the time when the DB2 Catalog and DBD were believed to be consistent.
72. A copy of the log for this period so that additional DSN1LOGP reports can be produced for problem determination.
73. The location used in the alias entry in SYSIBM.SYSTABLES.
74. The contents of the SYSIBM.LOCATIONS table.
75. The AUTHORIZATION ID, PLANNAME, and LUNAME of the thread used to select the MODENAME in the SYSIBM.MODESELECT table.
76. The contents of the SYSIBM.LOCATIONS table and the SYSIBM.MODESELECT.
77. Listing of the contents of SYSIBM.LUMODES in the communications database.
78. System log printout (obtained via MVS WRITELOG operator command) showing related VSAM/media manager messages that may appear there (usually needed only if you have a message or reason code that indicates a media manager services problem occurred).
79. A hexadecimal print of the first record of the failing DBRM.
80. The listing from the precompile job that generated the DBRM.
81. Output from the BIND attempt.
82. IMS or CICS trace and any dumps.
83. Statistics Class 4 trace record for the LUWID associated with the failing request.
84. All dumps created by this failure.
85. Job output
86. Print the contents of the SYSIBM.IPNAMES table.
87. Print the contents of the SYSIBM.IPLIST table.

Resource types

Table 2. Resource Types

TYPE Code	Type of Resource	Name, Content, Format
00000100	Database	DB
00000200	Table space	DB.SP
00000201	Index space	DB.SP
00000202	Table space	RD.DB.TS
00000205	Compression Dictionary	DB.SP
00000210	Partition	DB.SP.PT
00000220	Data set	DSN
00000230	Temporary file	SZ
00000240	Database procedure	DBP
00000300	Page	DB.SP.PG
00000301	Index minipage	DB.SP.PG.MP
00000302	Table space page	DB.SP.PG
00000303	Index space page	DB.SP.PG
00000304	Table space RID	DB.SP.RID
00000305	Index access/table space RID	DB.SP.RID
00000306	Index access/table space page	DB.SP.PG
00000307	Index space EOF	DB.SP.01
00000400	ICF catalog	IC
00000401	Authorization function	
00000402	Security Server	SAF/RACF return/reason codes
00000500	Storage group	SG
00000600	EDM pool space	
00000602	EDM DBD Space	
00000603	EDM DYNAMIC STATEMENT Space	
00000700	Buffer pool space	BP
00000701	Group buffer pool	GBP
00000800	Plan	PL
00000801	Package	COLLECTION. PACKAGE. CONTOKEN
00000802	BINDLOCK	BINDLOCK
00000900	32KB data area	
00000901	Sort storage	
00000903	Hash anchor	DB.SP.PG.AI
00000904	RIDLIST storage	
00000905	IRLM storage	
00000906	DB2	MEMBER
00000907	Data Space	MEMBER
00000908	Basic Floating Point Extensions Facility	
00000909	Extended Time-of-Day (TOD) Clock	
0000090A	XML storage	
00000A00	Table	RD.CR.TB
00000A10	Alias	RELDEP. OWNER. ALIAS
00000A11	Distinct type	SC.DT
00000A12	User-defined function	SC.SN
00000A13	Stored procedure	SC.SN
00000A14	Sequence	SC.SQ
00000B00	View	RD.CR.VW
00000C00	Index	RD.CR.IX
00000C01	Index	CR.IX
00000D00	DBID/OBID	RD.DI.OI
00000D01	DBID/OBID	DI.OI

Table 2. Resource Types (continued)

TYPE Code	Type of Resource	Name, Content, Format
00000D02	OBID	OI
00000E00	SU limit exceeded	CN
00000F00	Auxiliary column	DI.OI. ROWID. COLN
00000F01	LOB lock	DIX.PIX. ROWID. VRSN
00001000	DDF	LOCATION
00001001	System conversation	LU.MODE. RTNCD. FDBK2. RCPRI. RCSEC. SENSE
00001002	Agent conversation	LU.MODE. RTNCD. FDBK2. RCPRI. RCSEC. SENSE
00001003	CNOS processing	LU. MODE. RTNCD. FDBK2. RCPRI. RCSEC. SENSE
00001004	CDB (Communication database)	LOCATION. AUTHORIZATION ID. PL
00001005	DB access agent	LOCATION
00001007	TCP/IP domain name	LINKNAME. DOMAIN. ERRNO
00001008	TCP/IP service name	LOCATION. SERVICE. ERRNO
00001102	Bootstrap data set (BSDS)	MEMBER
00002000	Table space CS-claim class	DB.SP
00002001	Table space RR-claim class	DB.SP
00002002	Table space write-claim class	DB.SP
00002003	Index space CS-claim class	DB.SP
00002004	Index space RR-claim class	DB.SP
00002005	Index space write-claim class	DB.SP
00002006	Table space partition CS-claim class	DB.SP.PT
00002007	Table space partition RR-claim class	DB.SP.PT
00002008	Table space partition write-claim class	DB.SP.PT
00002009	Index space partition CS-claim class	DB.SP.PT
00002010	Index space partition RR-claim class	DB.SP.PT
00002011	Index space partition Write-claim class	DB.SP.PT
00002100	Table space DBET entry	DB.SP
00002101	Index space DBET entry	DB.SP
00002102	Table space partition DBET entry	DB.SP.PT
00002103	Index space partition DBET entry	DB.SP.PT
00002104	DBET hash chain lock timeout	INTERNAL LOCK NN
00002105	Logical partition DBET entry	DB.SP.PT
00002200	Routine Parameter Storage	DB.SP.PT
00002300	ICSF encryption and decryption facilities	
00003000	Code (release maintenance_level or system parameter)	REL, APAR, ZPARM
00003002	Number of Stored Procedures	
A10X	Alias	RD.CR.AL

The following table contains the definitions for the acronyms provided for the Name, Content, Format column of Table 2 on page 662.

Table 3. Definitions for resource type name, content, and format.

Where	Stands for
AI	Hash anchor ID
ALIAS	Alias owner
APAR	APAR number

Table 3. Definitions for resource type name, content, and format. (continued)

Where	Stands for
AUTHORIZATION ID	DB2 authorization identifier
BP	Buffer pool identifier
CN	Column name
COLLECTION	Collection-ID of the package
COLN	Column number within the base table of the LOB column that has been marked invalid.
CONTOKEN	Consistency token of the package
CR	Creator of the object
DB	Database name
DBP	Database procedure name
DI	DBID in decimal of resource
DIX	DBID in hexadecimal
DSN	Data set name
FDBK2	VTAM secondary return code
GBP	Group buffer pool name
IC	ICF catalog alias name
IX	Index name
LOCATION	Location in which the specified resource is not available
LU	Logical unit name
MEMBER	Group member name
MODE	Logical unit mode name
MP	Hexadecimal mini-page number
NN	decimal number of the hash chain
OI	OBID in decimal of resource
OWNER	Alias owner
PACKAGE	Package identifier
PG	Hexadecimal page number
PIX	PSID in hexadecimal
PL	Plan identifier
PT	Decimal partition number
RCPRI	APPC primary return code
RCSEC	APPC secondary return code
RD	DB2 release dependency mark
REL	DB2 release name
RELDEP	DB2 release dependency mark
RID	Record Identifier
RTNCD	VTAM primary return code
SENSE	SNA sense codename
SG	Storage group name
SP	Space name
SQ	Sequence name
SZ	Temporary file page size
TB	Table name
TS	Table space name
VW	View name

Note:

- DB can also refer to DI. SP, TS, and TB can also refer to OI. DB2 sometimes does not have access to resource names, and uses DBID or OBID instead.
- RD can be represented as a white space or blank to indicate no release dependency, or Y or N to indicate whether it is IBMREQD.

ABEND codes associated with DSN1COPY misuse

The following table represents the possible combinations of ABEND reason codes, CSECT-names, and ERQUAL's that can result from the improper use of DSN1COPY. An improper use includes, but is not limited to:

- Specifying the wrong OBIDXLAT values
- Not restoring all the datasets of the table space or index space to the same level
- Restoring datasets to a level that does not match the definition in the catalog and directory
- Restoring one table space type to a different type (e.g. segmented to simple).

This table is not meant to be exhaustive, but contains the more common codes, or those that are likely to have been caused by DSN1COPY misuse. However, this does not eliminate the possibility that the occurrence of these ABEND codes can be caused by other errors, such as an inconsistent DBD, or a DB2 internal error. The numbers under 'Explanation' refer to the notes following the table.

Table 4. ABEND Reason Codes Associated with DSN1COPY Use

ABEND Reason Code	CSECT-name: ERQUAL	Explanation
00C90101	DSNICKEY:5005	8
	DSNICKEY:5006	8
	DSNICSEL:5003	1
	DSNICSEL:5008	3
	DSNICUMW:5004	4
	DSNICUMW:5005	2
	DSNIDEFR:5005	13
	DSNIDLOD:501B	6
	DSNIIMSI:5006	3
	DSNIIMSI:5008	3
	DSNIIMSI:500A	3
	DSNIISF:5001	3
	DSNILPG:5013	6
	DSNIMOFR:500B	1
	DSNIMOFR:500C	1
	DSNIMRST:5001	1
	DSNIMRST:5002	1
	DSNIMRST:5003	1
	DSNIMSMS:5004	2
	DSNINXTP:5008	6
	DSNIOSEL:5002	2
	DSNIOSEL:5003	2
	DSNIOSEL:53DD	3
	DSNIOSEL:53DE	3
	DSNIPOCR:5005	2
	DSNIPOCR:5007	2
	DSNIPOCR:5009	2
	DSNIPOCR:500B	2
	DSNIPOCR:500D	2
	DSNIPOCR:500F	2
	DSNIPTYM:5006	6
	DSNIRECU:5002	2
	DSNIREDR:5007	1
	DSNIREDR:5008	1
	DSNIREDR:5009	1
	DSNIRELF:5011	2

Table 4. ABEND Reason Codes Associated with DSN1COPY Use (continued)

ABEND Reason Code	CSECT-name: ERQUAL	Explanation
	DSNIRELK:5001	1
	DSNIRLPG:500B	2
	DSNIRLPG:500C	1
	DSNIRLPG:5011	2
	DSNIRNXT:53D3	3
	DSNIRNXT:53DD	3
	DSNIRNXT:53DE	3
	DSNISEGD:5002	2
	DSNISEGD:5005	6
	DSNISEGF:5002	2
	DSNISEGR:500A	6
	DSNISEGR:500C	2
	DSNISEL:5006	1
	DSNISEL:5007	1
	DSNISEL:500A	3
	DSNISEL:500D	3
	DSNISESE:5001	1
	DSNISESE:5002	1
	DSNISESE:5003	1
	DSNISESR:5001	2
	DSNISESR:5002	1
	DSNISESR:5004	1
	DSNISESR:5005	1
	DSNISFS:53DD	3
	DSNISFS:53DE	3
	DSNISFX:5008	2
	DSNISGNS:5003	2
	DSNISGNS:5004	2
	DSNISGSS:5001	1
	DSNISMRS:5001	1
	DSNISRID:53DD	3
	DSNISRID:53DE	3
	DSNISRTI:5004	1
	DSNIWLPG:5003	6
	DSNKCKIS:5003	11
	DSNKFIND:5001	11
	DSNKFPNT:5003	12
	DSNKLVLN:5006	11
	DSNKLVLN:5007	11
	DSNKMDLE:5005	10
	DSNKMDLE:5007	10
	DSNKNEXT:5001	11
	DSNKNEXT:5004	11
	DSNKPOCI:5004	11
00C90105	DSNIBROK:0CA4	6
00C90206	DSNIIDIS:5002	4
00C90210	DSNKFIND:5002	11
	DSNKFIND:5005	11
	DSNKFIND:500B	11

Table 4. ABEND Reason Codes Associated with DSN1COPY Use (continued)

ABEND Reason Code	CSECT-name: ERQUAL	Explanation
00C90212	DSNKDNEX:5005	11
00C90213	DSNIRSET:500C	2
00C90214	DSNKISRT:5005	8
	DSNKISRT:5006	11
	DSNKLOD:500F	8
	DSNKLOD:5011	11
00C90215	DSNKFIND:5006	9
	DSNKFIND:5007	9
00C90216	DSNIRFNX:5008	1
00C90218	DSNIDLOD:5014	6
	DSNISEGD:5007	6
	DSNISGSC:5001	6
	DSNISNPG:5004	6
00C90219	DSNIDLOD:5015	2
	DSNISEGD:5008	2
	DSNISGSC:5002	2
	DSNISNPG:5005	2
00C9021A	DSNIRFNX:500A	2
	DSNIRFNX:500B	2
	DSNISFS:5007	2
	DSNISFS:5008	2
00C90221	DSNISGAU:5007	6
00C9021C	DSNIRFNX:5006	6
	DSNISNPG:5007	6

Explanations

The following explanations are groups of inconsistencies that relate to the ABEND codes, separated by user table spaces and user index spaces. Within each group are examples of specific inconsistencies representing each group.

User Table Spaces

1. Record or field length in the data does not match the length in the OBD
   - Either the fixed lengths do not match, or the variable length of the record is greater than the maximum allowed in the OBD
2. Record OBID in the data does not agree with the OBD
   - OBD corresponding to the OBID from the data cannot be found in the DBD, or the OBD in the DBD is of a different type.
3. Number of columns in the data row does not match the OBD definition

- An ALTER ADD COLUMN was performed, but the copy of the data is downlevel with respect to the ALTER ADD DBD.

4. Index does not match the data
   - Either the index entry does not exist for the data record, or the index entry points to a missing or bad data record.
5. Data in one table space is inconsistent with related data in another table space
   - Referential integrity; parent and dependent are in different table spaces.
6. Table space definition in the OBD is inconsistent with the data
   - SEGSIZEs do not match
   - Table space type is different (simple restored into segmented)
   - Segment anchor in the OBD points to an invalid first segment (segment not allocated).
7. Table space is inconsistent within itself
   - Not all datasets are restored to the same level
   - Broken segment chains
   - Base and overflow records are not consistent.

User Index Spaces

8. Total key length in the index page or entry does not match the key length in the OBD
   - Index data restored with the wrong index or downlevel index.
9. Key length derived from data passed into the Index Manager does not match the length in the index page or entry
   - Data or index restored to inconsistent states.
10. Index OBID in the index does not agree with the OBD
    - OBD corresponding to the OBID from the index cannot be found in the DBD, or the OBD in the DBD is of a different type.
11. Index space definition in the OBD is inconsistent with the index data
    - Inconsistent subpage directory
    - Cannot find the key in the partition high-key directory
    - Keys not found.
12. Index space is inconsistent within itself
    - Not all datasets are restored to the same level
    - Index entry not connected to the root page.
13. Log RBA is unavailable for the associated table space or index space.

Appendix B. New, changed, and deleted messages

This section lists new, changed, and deleted messages to support DB2 Version 8. This list does not include editorial changes or updates to book references. Changes are categorized by whether the messages are new, changed, or deleted:

- "New messages" lists new messages.
- "Changed messages" on page 677 lists changed messages.
- "Deleted messages" on page 682 lists deleted messages.

New messages

This section lists new messages to support DB2 Version 8.

DSNA... messages

- DSNA301I
- DSNA302I
- DSNA303I
- DSNA304I
- DSNA305I
- DSNA306I
- DSNA307I
- DSNA308I
- DSNA309I
- DSNA310I
- DSNA311I
- DSNA312I
- DSNA313I
- DSNA314I
- DSNA315I
- DSNA316I
- DSNA317I
- DSNA318I
- DSNA319I
- DSNA320I
- DSNA601I
- DSNA602I
- DSNA603I
- DSNA604I
- DSNA605I
- DSNA607I
- DSNA610I
- DSNA611I
- DSNA612I
- DSNA613I
- DSNA614I

© Copyright IBM Corp. 1983, 2008

• DSNA615I
• DSNA616I
• DSNA617I
• DSNA618I
• DSNA620I
• DSNA621I
• DSNA622I
• DSNA623I
• DSNA624I
• DSNA625I
• DSNA626I
• DSNA629I
• DSNA634I
• DSNA635I
• DSNA636I
• DSNA637I
• DSNA638I
• DSNA639I
• DSNA640I
• DSNA643I
• DSNA645I
• DSNA646I
• DSNA650I
• DSNA651I
• DSNA652I
• DSNA653I
• DSNA654I
• DSNA655I
• DSNA656I
• DSNA657I
• DSNA659I
• DSNA660I
• DSNA670I
• DSNA671I
• DSNA672I
• DSNA673I
• DSNA674I
• DSNA675I
• DSNA676I
• DSNA677I
• DSNA678I
• DSNA679I
• DSNA680I
• DSNA681I
• DSNA682I

- DSNA683I
- DSNA684I
- DSNA690I
- DSNA691I
- DSNA692I
- DSNA693I
- DSNA694I

DSNB... messages

- DSNB260I
- DSNB357I
- DSNB539I
- DSNB541I
- DSNB610I
- DSNB802I
- DSNB815I
- DSNB816I

DSNE... messages

- DSNE345I
- DSNE346I
- DSNE136I

DSNG... messages

- DSNG004I
- DSNG005I
- DSNG006I
- DSNG007I

DSNH... messages

- DSNH091I
- DSNH330I
- DSNH331I
- DSNH332I
- DSNH526I
- DSNH527I
- DSNH902I
- DSNH4700I
- DSNH4701I
- DSNH4702I
- DSNH4760I
- DSNH4761I
- DSNH4764I
- DSNH4771I
- DSNH4784I
- DSNH4788I
- DSNH5011I
- DSNH20289I

DSNI... messages

- DSNI028I
- DSNI029I
- DSNI041I

DSNJ... messages

- DSNJ016E
- DSNJ017E
- DSNJ032I
- DSNJ128I
- DSNJ132I
- DSNJ138I
- DSNJ155I
- DSNJ156I
- DSNJ254I
- DSNJ260I
- DSNJ261I
- DSNJ266I
- DSNJ439I
- DSNJ440I
- DSNJ441I
- DSNJ708I
- DSNJ990I

DSNL... messages

- DSNL523I

DSNQ... messages

- DSNQ001I
- DSNQ002I
- DSNQ003I
- DSNQ004I
- DSNQ010I
- DSNQ011I
- DSNQ012I
- DSNQ020I
- DSNQ021I
- DSNQ030I
- DSNQ040I
- DSNQ060I
- DSNQ061I
- DSNQ070I
- DSNQ071I
- DSNQ072I
- DSNQ073I
- DSNQ074I
- DSNQ075I

- DSNQ080I
- DSNQ081I
- DSNQ082I

DSNR... messages

- DSNR045I
- DSNR047I
- DSNR048I
- DSNR050I
- DSNR051I
- DSNR052I
- DSNR053I
- DSNR054I
- DSNR055I
- DSNR056I
- DSNR057I
- DSNR058I

DSNS... messages

- DSNS001I
- DSNS002I

DSNT... messages

- DSNT108I
- DSNT109I
- DSNT110I
- DSNT222I
- DSNT318I
- DSNT386I
- DSNT387I
- DSNT435I
- DSNT520I
- DSNT527I
- DSNT528I
- DSNT532I
- DSNT533I
- DSNT534I
- DSNT544I
- DSNT545I
- DSNT546I
- DSNT547I
- DSNT550I
- DSNT551I
- DSNT552I
- DSNT570I
- DSNT571I
- DSNT787I

- DSNT815I
- DSNT816I
- DSNT818I

DSNU... messages

- DSNU021I
- DSNU113I
- DSNU033I
- DSNU083I
- DSNU118I
- DSNU237I
- DSNU283I
- DSNU284I
- DSNU326I
- DSNU334I
- DSNU430I
- DSNU431I
- DSNU432I
- DSNU448I
- DSNU516I
- DSNU556I
- DSNU631I
- DSNU635I
- DSNU688I
- DSNU689I
- DSNU790I
- DSNU807I
- DSNU932I
- DSNU1001I
- DSNU1011I
- DSNU1039I
- DSNU1045I
- DSNU1050I
- DSNU1126I
- DSNU1127I
- DSNU1128I
- DSNU1129I
- DSNU1130I
- DSNU1141I
- DSNU1142I
- DSNU1143I
- DSNU1144I
- DSNU1145I
- DSNU1146I
- DSNU1147I
- DSNU1148I

#
#
|
#
#
#

- DSNU1623I
- DSNU1624I
- DSNU1626I
- DSNU1627I
- DSNU1628I
- DSNU1629I
- DSNU1631I
- DSNU1632I
- DSNU1633I
- DSNU1634I
- DSNU1640I
- DSNU1635I
- DSNU3333I

DSNV... messages

- DSNV440I
- DSNV441I

DSNX... messages

- DSNX210I
- DSNX211I
- DSNX230I
- DSNX941I
- DSNX942I
- DSNX969I
- DSNX970I

DSNY... messages

- DSNY009I
- DSNY011I
- DSNY012I

DSNZ... messages

- DSNZ015I

DSN1... messages

- DSN1191I
- DSN1192I
- DSN1193I
- DSN1194I
- DSN1195I
- DSN1196I
- DSN1197I
- DSN1198I
- DSN1930I
- DSN1931I
- DSN1939I

DSN3... messages

- DSN3028I

- DSN3029I

DSN7... messages

- DSN7101I
- DSN7130I
- DSN7131I
- DSN7132I
- DSN7133I

DSN9... messages

- DSN9035I
- DSN9036I
- DSN9037I
- DSN9038I

Changed messages

This section lists updated messages to support DB2 Version 8.

DSNB... messages

- DSNB207I
- DSNB226I
- DSNB250E
- DSNB306I
- DSNB332I
- DSNB333I
- DSNB357I
- DSNB402I
- DSNB404I
- DSNB402I
- DSNB406I
- DSNB411I
- DSNB420I
- DSNB502I
- DSNB508I
- DSNB514I
- DSNB540I
- DSNB744I
- DSNB745I
- DSNB749I
- DSNB755I
- DSNB757I
- DSNB758I
- DSNB761I
- DSNB762I
- DSNB764I
- DSNB768I
- DSNB773I

- DSNB774I
- DSNB775I
- DSNB776I
- DSNB777I
- DSNB786I
- DSNB787I
- DSNB789I
- DSNB796I
- DSNB799I
- DSNB800I

DSNE... messages

- DSNE926E
- DSNE955I

DSNH... messages

- DSNH007I
- DSNH102I
- DSNH107I
- DSNH109I
- DSNH110I
- DSNH4001I
- DSNH5036I
- DSNH527I
- DSNH794I
- DSNH901I

DSNI... messages

- DSNI005I
- DSNI006I
- DSNI021I
- DSNI022I
- DSNI023I
- DSNI031I
- DSNI033I

DSNJ... messages

- DSNJ103I
- DSNJ113
- DSNJ115I
- DSNJ120I
- DSNJ128I
- DSNJ153E
- DSNJ216I
- DSNJ220I
- DSNJ225I
- DSNJ318I
- DSNJ329I

- DSNJ338I
- DSNJ351I
- DSNJ357E
- DSNJ371I
- DSNJ374I
- DSNJ439I
- DSNJ440I
- DSNJ991I

DSNL... messages

- DSNL004I
- DSNL021I
- DSNL044I
- DSNL045I
- DSNL084I
- DSNL092I
- DSNL093I
- DSNL453I
- DSNL512I
- DSNL515I
- DSNL519I

DSNP... messages

- DSNP010I
- DSNP012I
- DSNP031I

DSNR... messages

- DSNR004I
- DSNR025I

DSNT... messages

- DSNT253I
- DSNT255I
- DSNT361I
- DSNT389I
- DSNT392I
- DSNT394I
- DSNT396I
- DSNT397I
- DSNT409I
- DSNT429I
- DSNT438I
- DSNT440I
- DSNT448I
- DSNT482I
- DSNT703I
- DSNT706I

- DSNT724E
- DSNT806I
- DSNT815I

DSNU... messages

- DSNU003I
- DSNU014I
- DSNU016I
- DSNU017I
- DSNU047I
- DSNU060I
- DSNU068I
- DSNU083I
- DSNU112I
- DSNU180I
- DSNU182I
- DSNU185I
- DSNU251I
- DSNU252I
- DSNU273I
- DSNU297I
- DSNU299I
- DSNU303I
- DSNU304I
- DSNU314I
- DSNU320I
- DSNU326I
- DSNU331I
- DSNU334I
- DSNU393I
- DSNU404I
- DSNU407I
- DSNU423I
- DNSU434I
- DSNU440I
- DSNU441I
- DSNU501I
- DSNU506I
- DSNU547I
- DSNU556I
- DSNU557I
- DSNU559I
- DSNU568I
- DSNU570I
- DSNU590I
- DNSU628I

#
- DNSU630I

#
- DSNU688I

#
- DSNU689I

#
- DSNU743I

#
- DSNU7787I

#
- DSNU784I

#
- DSNU785I

|
- DSNU861I

#
- DSNU787I

#
- DSNU805I

|
- DNSU814I

|
- DSNU815I

#
- DSNU972I

#
- DNSU1001I

|
- DSNU1038I

#
- DSNU1042I

#
- DSNU1043I

#
- DSNU1127I

#
- DSNU1128I

#
- DSNU1141I

#
- DSNU1401I

#
- DSNU1402I

#
- DSNU1404I

#
- DSNU1408I

#
- DSNU1702I

#
- DSNU1703I

#
- DSNU1704I

#
- DSNU1706I

#
- DSNU1708I

#
- DSNU1710I

#
- DSNU1712I

#
- DSNU1714I

#
- DSNU1716I

#
- DSNU1718I

#
- DSNU1720I

#
- DSNU1722I

#
- DSNU1724I

#
- DSNU1725I

#
- DSNU1726I

#
- DSNU1727I

#
- DSNU1728I

#
- DSNU1730I

#
- DSNU1732I

#
- DSNU1734I

#
- DSNU1735I

DSNV... messages

- DSNV404I
- DNSV463I

DSNW... messages

- DSNW130I

DSNX... messages

- DSNX204I
- DSNX210I
- DSNX900E
- DSNX941I
- DSNX942I
- DSNX961I
- DSNX969I
- DSNX970I
- DSNX986I
- DSNX987I

DSNY... messages

- DSNY009I

DSN1... messages

- DSN1957I
- DSN1946I
- DSN1967I

DSN3... messages

- DSN3100I
- DSN3201I

DSN7... messages

- DSN7101I
- DSN7104I
- DSN7107I
- DSN7513I

Deleted messages

The following messages were deleted:

DSNB... messages

- DSNB230I
- DSNB231I
- DSNB403I
- DSNB405I
- DSNB430I
- DSNB431I
- DSNB454I
- DSNB503I
- DSNB505I

- DSNB509I
- DSNB513I
- DSNB514I
- DSNB515I
- DSNB518I
- DSNB523I
- DSNB525I
- DSNB528I
- DSNB535I
- DSNB537I
- DSNB541I
- DSNB606I
- DSNB607I
- DSNB608I
- DSNB610I
- DSNB611I
- DSNB779I

DSNH... messages

- DSNH085I
- DSNH527I

DSNI... messages

- DSNI035I

DSNJ... messages

- DSNJ252I

DSNR... messages

- DSNR043E
- DSNR044E

DSNT... messages

- DSNT441I
- DSNT522I
- DSNT524I
- DSNT525I

DSNU... messages

- DSNU785I
- DSNU787I
- DSNU1145I

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

© Copyright IBM Corp. 1983, 2008

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Programming interface information

This book is intended to help you interpret messages issued by DB2 Universal Database for z/OS (DB2 UDB for z/OS).

This book also documents General-use Programming Interface and Associated Guidance Information provided by DB2 UDB for z/OS.

General-use programming interfaces allow the customer to write programs that obtain the services of DB2 UDB for z/OS.

General-use Programming Interface and Associated Guidance Information is identified where it occurs by an introductory statement to a chapter or section.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, or other countries, or both:

AD/Cycle
AIX
APL2
AS/400
BookManager
CICS
CICS/ESA
CICS/MVS
COBOL/370
C/370
DATABASE 2
DataHub
DataPropagator
DB2
DB2 Connect
DB2 Universal Database
DFSMS
DFSMSdfp
DFSMSdss
DFSMShsm
DFSMS/MVS
DFSORT
Distributed Relational Database Architecture
DRDA
DXT
eNetwork
Enterprise System/3090
Enterprise System/9000
ES/3090
Extended Services
IBM
IMS
IMS/ESA
MVS/DFP
MVS/ESA
MVS/XA
Net.Data
OS/2
OS/390
OS/400
Parallel Sysplex
PR/SM
QMF
RACF
RMF
SQL/DS
System/370
System/390
S/370
VTAM
z/OS
3090

Tivoli™ and NetView are trademarks of Tivoli Systems Inc. in the United States, or other countries, or both.

The following terms are trademarks of other companies as follows:

- Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.
- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft™ Corporation, Inc. in the United States and/or other countries.

Other company, product, and service names may be trademarks or service marks of others.

Glossary

The following terms and abbreviations are defined as they are used in the DB2 library.

A

abend. Abnormal end of task.

abend reason code. A 4-byte hexadecimal code that uniquely identifies a problem with DB2.

abnormal end of task (abend). Termination of a task, job, or subsystem because of an error condition that recovery facilities cannot resolve during execution.

access method services. The facility that is used to define and reproduce VSAM key-sequenced data sets.

access path. The path that is used to locate data that is specified in SQL statements. An access path can be indexed or sequential.

active log. The portion of the DB2 log to which log records are written as they are generated. The active log always contains the most recent log records, whereas the archive log holds those records that are older and no longer fit on the active log.

active member state. A state of a member of a data sharing group. The cross-system coupling facility identifies each active member with a group and associates the member with a particular task, address space, and z/OS system. A member that is not active has either a failed member state or a quiesced member state.

address space. A range of virtual storage pages that is identified by a number (ASID) and a collection of segment and page tables that map the virtual pages to real pages of the computer's memory.

address space connection. The result of connecting an allied address space to DB2. Each address space that contains a task that is connected to DB2 has exactly one address space connection, even though more than one task control block (TCB) can be present. See also *allied address space* and *task control block*.

| **address space identifier (ASID).** A unique
| system-assigned identifier for and address space.

administrative authority. A set of related privileges that DB2 defines. When you grant one of the administrative authorities to a person's ID, the person has all of the privileges that are associated with that administrative authority.

after trigger. A trigger that is defined with the trigger activation time AFTER.

agent. As used in DB2, the structure that associates all processes that are involved in a DB2 unit of work. An *allied agent* is generally synonymous with an *allied thread*. *System agents* are units of work that process tasks that are independent of the allied agent, such as prefetch processing, deferred writes, and service tasks.

aggregate function. An operation that derives its
result by using values from one or more rows. Contrast
with *scalar function*.

alias. An alternative name that can be used in SQL statements to refer to a table or view in the same or a remote DB2 subsystem.

allied address space. An area of storage that is external to DB2 and that is connected to DB2. An allied address space is capable of requesting DB2 services.

allied thread. A thread that originates at the local DB2 subsystem and that can access data at a remote DB2 subsystem.

allocated cursor. A cursor that is defined for stored procedure result sets by using the SQL ALLOCATE CURSOR statement.

already verified. An LU 6.2 security option that allows DB2 to provide the user's verified authorization ID when allocating a conversation. With this option, the user is not validated by the partner DB2 subsystem.

| **ambiguous cursor.** A database cursor that is in a plan
| or package that contains either PREPARE or EXECUTE
| IMMEDIATE SQL statements, and for which the
| following statements are true: the cursor is not defined
| with the FOR READ ONLY clause or the FOR UPDATE
| OF clause; the cursor is not defined on a read-only
| result table; the cursor is not the target of a WHERE
| CURRENT clause on an SQL UPDATE or DELETE
| statement.

American National Standards Institute (ANSI). An organization consisting of producers, consumers, and general interest groups, that establishes the procedures by which accredited organizations create and maintain voluntary industry standards in the United States.

ANSI. American National Standards Institute.

APAR. Authorized program analysis report.

APAR fix corrective service. A temporary correction of an IBM software defect. The correction is temporary,

© Copyright IBM Corp. 1983, 2008

because it is usually replaced at a later date by a more permanent correction, such as a program temporary fix (PTF).

APF. Authorized program facility.

API. Application programming interface.

APPL. A VTAM® network definition statement that is used to define DB2 to VTAM as an application program that uses SNA LU 6.2 protocols.

application. A program or set of programs that performs a task; for example, a payroll application.

application-directed connection. A connection that an application manages using the SQL CONNECT statement.

application plan. The control structure that is produced during the bind process. DB2 uses the application plan to process SQL statements that it encounters during statement execution.

application process. The unit to which resources and locks are allocated. An application process involves the execution of one or more programs.

application programming interface (API). A functional interface that is supplied by the operating system or by a separately orderable licensed program that allows an application program that is written in a high-level language to use specific data or functions of the operating system or licensed program.

application requester. The component on a remote system that generates DRDA® requests for data on behalf of an application. An application requester accesses a DB2 database server using the DRDA application-directed protocol.

application server. The target of a request from a remote application. In the DB2 environment, the application server function is provided by the distributed data facility and is used to access DB2 data from remote applications.

archive log. The portion of the DB2 log that contains log records that have been copied from the active log.

ASCII. An encoding scheme that is used to represent strings in many environments, typically on PCs and workstations. Contrast with *EBCDIC* and *Unicode*.

ASID. Address space identifier.

attachment facility. An interface between DB2 and TSO, IMS, CICS, or batch address spaces. An attachment facility allows application programs to access DB2.

attribute. A characteristic of an entity. For example, in database design, the phone number of an employee is one of that employee's attributes.

authorization ID. A string that can be verified for connection to DB2 and to which a set of privileges is allowed. It can represent an individual, an organizational group, or a function, but DB2 does not determine this representation.

authorized program analysis report (APAR). A report of a problem that is caused by a suspected defect in a current release of an IBM supplied program.

authorized program facility (APF). A facility that permits the identification of programs that are authorized to use restricted functions.

automatic query rewrite. A process that examines an SQL statement that refers to one or more base tables, and, if appropriate, rewrites the query so that it performs better. This process can also determine whether to rewrite a query so that it refers to one or more materialized query tables that are derived from the source tables.

auxiliary index. An index on an auxiliary table in which each index entry refers to a LOB.

auxiliary table. A table that stores columns outside the table in which they are defined. Contrast with *base table*.

B

backout. The process of undoing uncommitted changes that an application process made. This might be necessary in the event of a failure on the part of an application process, or as a result of a deadlock situation.

backward log recovery. The fourth and final phase of restart processing during which DB2 scans the log in a backward direction to apply UNDO log records for all aborted changes.

base table. (1) A table that is created by the SQL CREATE TABLE statement and that holds persistent data. Contrast with *result table* and *temporary table*.

(2) A table containing a LOB column definition. The actual LOB column data is not stored with the base table. The base table contains a row identifier for each row and an indicator column for each of its LOB columns. Contrast with *auxiliary table*.

base table space. A table space that contains base tables.

basic predicate. A predicate that compares two values.

basic sequential access method (BSAM). An access method for storing or retrieving data blocks in a continuous sequence, using either a sequential-access or a direct-access device.

batch message processing program. In IMS, an application program that can perform batch-type processing online and can access the IMS input and output message queues.

before trigger. A trigger that is defined with the trigger activation time BEFORE.

binary integer. A basic data type that can be further classified as small integer or large integer.

binary large object (BLOB). A sequence of bytes in which the size of the value ranges from 0 bytes to 2 GB–1. Such a string has a CCSID value of 65535.

binary string. A sequence of bytes that is not associated with a CCSID. For example, the BLOB data type is a binary string.

bind. The process by which the output from the SQL precompiler is converted to a usable control structure, often called an access plan, application plan, or package. During this process, access paths to the data are selected and some authorization checking is performed. The types of bind are:

automatic bind. (More correctly, *automatic rebind*) A process by which SQL statements are bound automatically (without a user issuing a BIND command) when an application process begins execution and the bound application plan or package it requires is not valid.

dynamic bind. A process by which SQL statements are bound as they are entered.

incremental bind. A process by which SQL statements are bound during the execution of an application process.

static bind. A process by which SQL statements are bound after they have been precompiled. All static SQL statements are prepared for execution at the same time.

bit data. Data that is character type CHAR or VARCHAR and has a CCSID value of 65535.

BLOB. Binary large object.

block fetch. A capability in which DB2 can retrieve, or fetch, a large set of rows together. Using block fetch can significantly reduce the number of messages that are being sent across the network. Block fetch applies only to cursors that do not update data.

BMP. Batch Message Processing (IMS). See *batch message processing program*.

bootstrap data set (BSDS). A VSAM data set that contains name and status information for DB2, as well as RBA range specifications, for all active and archive log data sets. It also contains passwords for the DB2 directory and catalog, and lists of conditional restart and checkpoint records.

BSAM. Basic sequential access method.

BSDS. Bootstrap data set.

buffer pool. Main storage that is reserved to satisfy the buffering requirements for one or more table spaces or indexes.

built-in data type. A data type that IBM supplies. Among the built-in data types for DB2 UDB for z/OS are string, numeric, ROWID, and datetime. Contrast with *distinct type*.

built-in function. A function that DB2 supplies. Contrast with *user-defined function*.

business dimension. A category of data, such as products or time periods, that an organization might want to analyze.

C

cache structure. A coupling facility structure that stores data that can be available to all members of a Sysplex. A DB2 data sharing group uses cache structures as group buffer pools.

CAF. Call attachment facility.

call attachment facility (CAF). A DB2 attachment facility for application programs that run in TSO or z/OS batch. The CAF is an alternative to the DSN command processor and provides greater control over the execution environment.

call-level interface (CLI). A callable application programming interface (API) for database access, which is an alternative to using embedded SQL. In contrast to embedded SQL, DB2 ODBC (which is based on the CLI architecture) does not require the user to precompile or bind applications, but instead provides a standard set of functions to process SQL statements and related services at run time.

cascade delete. The way in which DB2 enforces referential constraints when it deletes all descendent rows of a deleted parent row.

CASE expression. An expression that is selected based on the evaluation of one or more conditions.

cast function. A function that is used to convert instances of a (source) data type into instances of a different (target) data type. In general, a cast function has the name of the target data type. It has one single argument whose type is the source data type; its return type is the target data type.

castout. The DB2 process of writing changed pages from a group buffer pool to disk.

castout owner. The DB2 member that is responsible for casting out a particular page set or partition.

catalog. In DB2, a collection of tables that contains descriptions of objects such as tables, views, and indexes.

catalog table. Any table in the DB2 catalog.

CCSID. Coded character set identifier.

CDB. Communications database.

CDRA. Character Data Representation Architecture.

CEC. Central electronic complex. See *central processor complex*.

central electronic complex (CEC). See *central processor complex*.

central processor (CP). The part of the computer that contains the sequencing and processing facilities for instruction execution, initial program load, and other machine operations.

central processor complex (CPC). A physical collection of hardware (such as an ES/3090™) that consists of main storage, one or more central processors, timers, and channels.

CFRM. Coupling facility resource management.

CFRM policy. A declaration by a z/OS administrator regarding the allocation rules for a coupling facility structure.

character conversion. The process of changing characters from one encoding scheme to another.

Character Data Representation Architecture (CDRA). An architecture that is used to achieve consistent representation, processing, and interchange of string data.

character large object (CLOB). A sequence of bytes representing single-byte characters or a mixture of single- and double-byte characters where the size of the value can be up to 2 GB–1. In general, character large object values are used whenever a character string might exceed the limits of the VARCHAR type.

character set. A defined set of characters.

character string. A sequence of bytes that represent bit data, single-byte characters, or a mixture of single-byte and multibyte characters.

check constraint. A user-defined constraint that specifies the values that specific columns of a base table can contain.

check integrity. The condition that exists when each row in a table conforms to the check constraints that are defined on that table. Maintaining check integrity requires DB2 to enforce check constraints on operations that add or change data.

check pending. A state of a table space or partition that prevents its use by some utilities and by some SQL statements because of rows that violate referential constraints, check constraints, or both.

checkpoint. A point at which DB2 records internal status information on the DB2 log; the recovery process uses this information if DB2 abnormally terminates.

child lock. For explicit hierarchical locking, a lock that is held on either a table, page, row, or a large object (LOB). Each child lock has a parent lock. See also *parent lock*.

CI. Control interval.

CICS. Represents (in this publication): CICS Transaction Server for z/OS: Customer Information Control System Transaction Server for z/OS.

CICS attachment facility. A DB2 subcomponent that uses the z/OS subsystem interface (SSI) and cross-storage linkage to process requests from CICS to DB2 and to coordinate resource commitment.

CIDF. Control interval definition field.

claim. A notification to DB2 that an object is being accessed. Claims prevent drains from occurring until the claim is released, which usually occurs at a commit point. Contrast with *drain*.

claim class. A specific type of object access that can be one of the following isolation levels:
Cursor stability (CS)
Repeatable read (RR)
Write

claim count. A count of the number of agents that are accessing an object.

class of service. A VTAM term for a list of routes through a network, arranged in an order of preference for their use.

class word. A single word that indicates the nature of a data attribute. For example, the class word PROJ indicates that the attribute identifies a project.

clause. In SQL, a distinct part of a statement, such as a SELECT clause or a WHERE clause.

CLI. Call- level interface.

client. See *requester*.

CLIST. Command list. A language for performing TSO tasks.

CLOB. Character large object.

closed application. An application that requires exclusive use of certain statements on certain DB2

objects, so that the objects are managed solely through the application's external interface.

CLPA. Create link pack area.

clustering index. An index that determines how rows are physically ordered (*clustered*) in a table space. If a clustering index on a partitioned table is not a partitioning index, the rows are ordered in cluster sequence within each data partition instead of spanning partitions. Prior to Version 8 of DB2 UDB for z/OS, the partitioning index was required to be the clustering index.

coded character set. A set of unambiguous rules that establish a character set and the one-to-one relationships between the characters of the set and their coded representations.

coded character set identifier (CCSID). A 16-bit number that uniquely identifies a coded representation of graphic characters. It designates an encoding scheme identifier and one or more pairs consisting of a character set identifier and an associated code page identifier.

code page. (1) A set of assignments of characters to code points. In EBCDIC, for example, the character 'A' is assigned code point X'C1' (2) , and character 'B' is assigned code point X'C2'. Within a code page, each code point has only one specific meaning.

code point. In CDRA, a unique bit pattern that represents a character in a code page.

code unit. The fundamental binary width in a computer architecture that is used for representing character data, such as 7 bits, 8 bits, 16 bits, or 32 bits. Depending on the character encoding form that is used, each code point in a coded character set can be represented internally by one or more code units.

coexistence. During migration, the period of time in which two releases exist in the same data sharing group.

cold start. A process by which DB2 restarts without processing any log records. Contrast with *warm start*.

collection. A group of packages that have the same qualifier.

column. The vertical component of a table. A column has a name and a particular data type (for example, character, decimal, or integer).

column function. See *aggregate function*.

"come from" checking. An LU 6.2 security option that defines a list of authorization IDs that are allowed to connect to DB2 from a partner LU.

command. A DB2 operator command or a DSN subcommand. A command is distinct from an SQL statement.

command prefix. A one- to eight-character command identifier. The command prefix distinguishes the command as belonging to an application or subsystem rather than to MVS.

command recognition character (CRC). A character that permits a z/OS console operator or an IMS subsystem user to route DB2 commands to specific DB2 subsystems.

command scope. The scope of command operation in a data sharing group. If a command has *member scope*, the command displays information only from the one member or affects only non-shared resources that are owned locally by that member. If a command has *group scope*, the command displays information from all members, affects non-shared resources that are owned locally by all members, displays information on sharable resources, or affects sharable resources.

commit. The operation that ends a unit of work by releasing locks so that the database changes that are made by that unit of work can be perceived by other processes.

commit point. A point in time when data is considered consistent.

committed phase. The second phase of the multisite update process that requests all participants to commit the effects of the logical unit of work.

common service area (CSA). In z/OS, a part of the common area that contains data areas that are addressable by all address spaces.

communications database (CDB). A set of tables in the DB2 catalog that are used to establish conversations with remote database management systems.

comparison operator. A token (such as =, >, or <) that is used to specify a relationship between two values.

composite key. An ordered set of key columns of the same table.

compression dictionary. The dictionary that controls the process of compression and decompression. This dictionary is created from the data in the table space or table space partition.

concurrency. The shared use of resources by more than one application process at the same time.

conditional restart. A DB2 restart that is directed by a user-defined conditional restart control record (CRCR).

connection. In SNA, the existence of a communication path between two partner LUs that allows information

to be exchanged (for example, two DB2 subsystems that are connected and communicating by way of a conversation).

connection context. In SQLJ, a Java™ object that represents a connection to a data source.

connection declaration clause. In SQLJ, a statement that declares a connection to a data source.

connection handle. The data object containing information that is associated with a connection that DB2 ODBC manages. This includes general status information, transaction status, and diagnostic information.

connection ID. An identifier that is supplied by the attachment facility and that is associated with a specific address space connection.

consistency token. A timestamp that is used to generate the version identifier for an application. See also *version*.

constant. A language element that specifies an unchanging value. Constants are classified as string constants or numeric constants. Contrast with *variable*.

constraint. A rule that limits the values that can be inserted, deleted, or updated in a table. See *referential constraint*, *check constraint*, and *unique constraint*.

context. The application's logical connection to the data source and associated internal DB2 ODBC connection information that allows the application to direct its operations to a data source. A DB2 ODBC context represents a DB2 thread.

contracting conversion. A process that occurs when the length of a converted string is smaller than that of the source string. For example, this process occurs when an EBCDIC mixed-data string that contains DBCS characters is converted to ASCII mixed data; the converted string is shorter because of the removal of the shift codes.

control interval (CI). A fixed-length area or disk in which VSAM stores records and creates distributed free space. Also, in a key-sequenced data set or file, the set of records that an entry in the sequence-set index record points to. The control interval is the unit of information that VSAM transmits to or from disk. A control interval always includes an integral number of physical records.

control interval definition field (CIDF). In VSAM, a field that is located in the 4 bytes at the end of each control interval; it describes the free space, if any, in the control interval.

conversation. Communication, which is based on LU 6.2 or Advanced Program-to-Program Communication (APPC), between an application and a remote transaction program over an SNA logical unit-to-logical unit (LU-LU) session that allows communication while processing a transaction.

coordinator. The system component that coordinates the commit or rollback of a unit of work that includes work that is done on one or more other systems.

copy pool. A named set of SMS storage groups that contains data that is to be copied collectively. A copy pool is an SMS construct that lets you define which storage groups are to be copied by using FlashCopy® functions. HSM determines which volumes belong to a copy pool.

copy target. A named set of SMS storage groups that are to be used as containers for copy pool volume copies. A copy target is an SMS construct that lets you define which storage groups are to be used as containers for volumes that are copied by using FlashCopy functions.

copy version. A point-in-time FlashCopy copy that is managed by HSM. Each copy pool has a version parameter that specifies how many copy versions are maintained on disk.

correlated columns. A relationship between the value of one column and the value of another column.

correlated subquery. A subquery (part of a WHERE or HAVING clause) that is applied to a row or group of rows of a table or view that is named in an outer subselect statement.

correlation ID. An identifier that is associated with a specific thread. In TSO, it is either an authorization ID or the job name.

correlation name. An identifier that designates a table, a view, or individual rows of a table or view within a single SQL statement. It can be defined in any FROM clause or in the first clause of an UPDATE or DELETE statement.

cost category. A category into which DB2 places cost estimates for SQL statements at the time the statement is bound. A cost estimate can be placed in either of the following cost categories:

- A: Indicates that DB2 had enough information to make a cost estimate without using default values.
- B: Indicates that some condition exists for which DB2 was forced to use default values for its estimate.

The cost category is externalized in the COST_CATEGORY column of the DSN_STATEMNT_TABLE when a statement is explained.

coupling facility. A special PR/SM™ LPAR logical partition that runs the coupling facility control program and provides high-speed caching, list processing, and locking functions in a Parallel Sysplex®.

coupling facility resource management. A component of z/OS that provides the services to manage coupling facility resources in a Parallel Sysplex. This management includes the enforcement of CFRM policies to ensure that the coupling facility and structure requirements are satisfied.

CP. Central processor.

CPC. Central processor complex.

C++ member. A data object or function in a structure, union, or class.

C++ member function. An operator or function that is declared as a member of a class. A member function has access to the private and protected data members and to the member functions of objects in its class. Member functions are also called methods.

C++ object. (1) A region of storage. An object is created when a variable is defined or a new function is invoked. (2) An instance of a class.

CRC. Command recognition character.

CRCR. Conditional restart control record. See also *conditional restart*.

create link pack area (CLPA). An option that is used during IPL to initialize the link pack pageable area.

created temporary table. A table that holds temporary data and is defined with the SQL statement CREATE GLOBAL TEMPORARY TABLE. Information about created temporary tables is stored in the DB2 catalog, so this kind of table is persistent and can be shared across application processes. Contrast with *declared temporary table*. See also *temporary table*.

cross-memory linkage. A method for invoking a program in a different address space. The invocation is synchronous with respect to the caller.

cross-system coupling facility (XCF). A component of z/OS that provides functions to support cooperation between authorized programs that run within a Sysplex.

cross-system extended services (XES). A set of z/OS services that allow multiple instances of an application or subsystem, running on different systems in a Sysplex environment, to implement high-performance, high-availability data sharing by using a coupling facility.

CS. Cursor stability.

CSA. Common service area.

CT. Cursor table.

current data. Data within a host structure that is current with (identical to) the data within the base table.

current SQL ID. An ID that, at a single point in time, holds the privileges that are exercised when certain dynamic SQL statements run. The current SQL ID can be a primary authorization ID or a secondary authorization ID.

current status rebuild. The second phase of restart processing during which the status of the subsystem is reconstructed from information on the log.

cursor. A named control structure that an application program uses to point to a single row or multiple rows within some ordered set of rows of a result table. A cursor can be used to retrieve, update, or delete rows from a result table.

cursor sensitivity. The degree to which database updates are visible to the subsequent FETCH statements in a cursor. A cursor can be sensitive to changes that are made with positioned update and delete statements specifying the name of that cursor. A cursor can also be sensitive to changes that are made with searched update or delete statements, or with cursors other than this cursor. These changes can be made by this application process or by another application process.

cursor stability (CS). The isolation level that provides maximum concurrency without the ability to read uncommitted data. With cursor stability, a unit of work holds locks only on its uncommitted changes and on the current row of each of its cursors.

cursor table (CT). The copy of the skeleton cursor table that is used by an executing application process.

cycle. A set of tables that can be ordered so that each table is a descendent of the one before it, and the first table is a descendent of the last table. A self-referencing table is a cycle with a single member.

D

DAD. See *Document access definition.*

disk. A direct-access storage device that records data magnetically.

database. A collection of tables, or a collection of table spaces and index spaces.

database access thread. A thread that accesses data at the local subsystem on behalf of a remote subsystem.

database administrator (DBA). An individual who is responsible for designing, developing, operating, safeguarding, maintaining, and using a database.

database alias. The name of the target server if different from the location name. The database alias name is used to provide the name of the database server as it is known to the network. When a database alias name is defined, the location name is used by the application to reference the server, but the database alias name is used to identify the database server to be accessed. Any fully qualified object names within any SQL statements are not modified and are sent unchanged to the database server.

database descriptor (DBD). An internal representation of a DB2 database definition, which reflects the data definition that is in the DB2 catalog. The objects that are defined in a database descriptor are table spaces, tables, indexes, index spaces, relationships, check constraints, and triggers. A DBD also contains information about accessing tables in the database.

database exception status. An indication that something is wrong with a database. All members of a data sharing group must know and share the exception status of databases.

database identifier (DBID). An internal identifier of the database.

database management system (DBMS). A software system that controls the creation, organization, and modification of a database and the access to the data that is stored within it.

database request module (DBRM). A data set member that is created by the DB2 precompiler and that contains information about SQL statements. DBRMs are used in the bind process.

database server. The target of a request from a local application or an intermediate database server. In the DB2 environment, the database server function is provided by the distributed data facility to access DB2 data from local applications, or from a remote database server that acts as an intermediate database server.

data currency. The state in which data that is retrieved into a host variable in your program is a copy of data in the base table.

data definition name (ddname). The name of a data definition (DD) statement that corresponds to a data control block containing the same name.

data dictionary. A repository of information about an organization's application programs, databases, logical data models, users, and authorizations. A data dictionary can be manual or automated.

data-driven business rules. Constraints on particular data values that exist as a result of requirements of the business.

Data Language/I (DL/I). The IMS data manipulation language; a common high-level interface between a user application and IMS.

data mart. A small data warehouse that applies to a single department or team. See also *data warehouse*.

data mining. The process of collecting critical business information from a data warehouse, correlating it, and uncovering associations, patterns, and trends.

data partition. A VSAM data set that is contained within a partitioned table space.

data-partitioned secondary index (DPSI). A secondary index that is partitioned. The index is partitioned according to the underlying data.

data sharing. The ability of two or more DB2 subsystems to directly access and change a single set of data.

data sharing group. A collection of one or more DB2 subsystems that directly access and change the same data while maintaining data integrity.

data sharing member. A DB2 subsystem that is assigned by XCF services to a data sharing group.

data source. A local or remote relational or non-relational data manager that is capable of supporting data access via an ODBC driver that supports the ODBC APIs. In the case of DB2 UDB for z/OS, the data sources are always relational database managers.

data space. In releases prior to DB2 UDB for z/OS, Version 8, a range of up to 2 GB of contiguous virtual storage addresses that a program can directly manipulate. Unlike an address space, a data space can hold only data; it does not contain common areas, system data, or programs.

data type. An attribute of columns, literals, host variables, special registers, and the results of functions and expressions.

data warehouse. A system that provides critical business information to an organization. The data warehouse system cleanses the data for accuracy and currency, and then presents the data to decision makers so that they can interpret and use it effectively and efficiently.

date. A three-part value that designates a day, month, and year.

date duration. A decimal integer that represents a number of years, months, and days.

datetime value. A value of the data type DATE, TIME, or TIMESTAMP.

DBA. Database administrator.

DBCLOB. Double-byte character large object.

DBCS. Double-byte character set.

DBD. Database descriptor.

DBID. Database identifier.

DBMS. Database management system.

DBRM. Database request module.

DB2 catalog. Tables that are maintained by DB2 and contain descriptions of DB2 objects, such as tables, views, and indexes.

DB2 command. An instruction to the DB2 subsystem that a user enters to start or stop DB2, to display information on current users, to start or stop databases, to display information on the status of databases, and so on.

DB2 for VSE & VM. The IBM DB2 relational database management system for the VSE and VM operating systems.

DB2I. DB2 Interactive.

DB2 Interactive (DB2I). The DB2 facility that provides for the execution of SQL statements, DB2 (operator) commands, programmer commands, and utility invocation.

DB2I Kanji Feature. The tape that contains the panels and jobs that allow a site to display DB2I panels in Kanji.

DB2 PM. DB2 Performance Monitor.

DB2 thread. The DB2 structure that describes an application's connection, traces its progress, processes resource functions, and delimits its accessibility to DB2 resources and services.

DCLGEN. Declarations generator.

DDF. Distributed data facility.

ddname. Data definition name.

deadlock. Unresolvable contention for the use of a resource, such as a table or an index.

declarations generator (DCLGEN). A subcomponent of DB2 that generates SQL table declarations and COBOL, C, or PL/I data structure declarations that conform to the table. The declarations are generated from DB2 system catalog information. DCLGEN is also a DSN subcommand.

declared temporary table. A table that holds temporary data and is defined with the SQL statement DECLARE GLOBAL TEMPORARY TABLE. Information about declared temporary tables is not stored in the DB2 catalog, so this kind of table is not persistent and can be used only by the application process that issued the DECLARE statement. Contrast with *created temporary table*. See also *temporary table*.

default value. A predetermined value, attribute, or option that is assumed when no other is explicitly specified.

deferred embedded SQL. SQL statements that are neither fully static nor fully dynamic. Like static statements, they are embedded within an application, but like dynamic statements, they are prepared during the execution of the application.

deferred write. The process of asynchronously writing changed data pages to disk.

degree of parallelism. The number of concurrently executed operations that are initiated to process a query.

delete-connected. A table that is a dependent of table P or a dependent of a table to which delete operations from table P cascade.

delete hole. The location on which a cursor is positioned when a row in a result table is refetched and the row no longer exists on the base table, because another cursor deleted the row between the time the cursor first included the row in the result table and the time the cursor tried to refetch it.

delete rule. The rule that tells DB2 what to do to a dependent row when a parent row is deleted. For each relationship, the rule might be CASCADE, RESTRICT, SET NULL, or NO ACTION.

delete trigger. A trigger that is defined with the triggering SQL operation DELETE.

delimited identifier. A sequence of characters that are enclosed within double quotation marks ("). The sequence must consist of a letter followed by zero or more characters, each of which is a letter, digit, or the underscore character (_).

delimiter token. A string constant, a delimited identifier, an operator symbol, or any of the special characters that are shown in DB2 syntax diagrams.

denormalization. A key step in the task of building a physical relational database design. Denormalization is the intentional duplication of columns in multiple tables, and the consequence is increased data redundancy. Denormalization is sometimes necessary to minimize performance problems. Contrast with *normalization*.

dependent. An object (row, table, or table space) that has at least one parent. The object is also said to be a dependent (row, table, or table space) of its parent. See also *parent row*, *parent table*, *parent table space*.

dependent row. A row that contains a foreign key that matches the value of a primary key in the parent row.

dependent table. A table that is a dependent in at least one referential constraint.

DES-based authenticator. An authenticator that is generated using the DES algorithm.

descendent. An object that is a dependent of an object or is the dependent of a descendent of an object.

descendent row. A row that is dependent on another row, or a row that is a descendent of a dependent row.

descendent table. A table that is a dependent of another table, or a table that is a descendent of a dependent table.

deterministic function. A user-defined function whose result is dependent on the values of the input arguments. That is, successive invocations with the same input values produce the same answer. Sometimes referred to as a *not-variant* function. Contrast this with an *nondeterministic function* (sometimes called a *variant function*), which might not always produce the same result for the same inputs.

DFP. Data Facility Product (in z/OS).

DFSMS. Data Facility Storage Management Subsystem (in z/OS). Also called *Storage Management Subsystem (SMS)*.

| **DFSMSdss™.** The data set services (dss) component of
| DFSMS (in z/OS).

| **DFSMShsm™.** The hierarchical storage manager (hsm)
| component of DFSMS (in z/OS).

dimension. A data category such as time, products, or markets. The elements of a dimension are referred to as members. Dimensions offer a very concise, intuitive way of organizing and selecting data for retrieval, exploration, and analysis. See also *dimension table*.

dimension table. The representation of a dimension in a star schema. Each row in a dimension table represents all of the attributes for a particular member of the dimension. See also *dimension*, *star schema*, and *star join*.

directory. The DB2 system database that contains internal objects such as database descriptors and skeleton cursor tables.

distinct predicate. In SQL, a predicate that ensures
that two row values are not equal, and that both row
values are not null.

distinct type. A user-defined data type that is internally represented as an existing type (its source type), but is considered to be a separate and incompatible type for semantic purposes.

distributed data. Data that resides on a DBMS other than the local system.

distributed data facility (DDF). A set of DB2 components through which DB2 communicates with another relational database management system.

Distributed Relational Database Architecture™ (DRDA). A connection protocol for distributed relational database processing that is used by IBM's relational database products. DRDA includes protocols for communication between an application and a remote relational database management system, and for communication between relational database management systems. See also *DRDA access.*

DL/I. Data Language/I.

DNS. Domain name server.

| **document access definition (DAD).** Used to define
| the indexing scheme for an XML column or the
| mapping scheme of an XML collection. It can be used
| to enable an XML Extender column of an XML
| collection, which is XML formatted.

domain. The set of valid values for an attribute.

domain name. The name by which TCP/IP applications refer to a TCP/IP host within a TCP/IP network.

domain name server (DNS). A special TCP/IP network server that manages a distributed directory that is used to map TCP/IP host names to IP addresses.

double-byte character large object (DBCLOB). A sequence of bytes representing double-byte characters where the size of the values can be up to 2 GB. In general, DBCLOB values are used whenever a double-byte character string might exceed the limits of the VARGRAPHIC type.

double-byte character set (DBCS). A set of characters, which are used by national languages such as Japanese and Chinese, that have more symbols than can be represented by a single byte. Each character is 2 bytes in length. Contrast with *single-byte character set* and *multibyte character set*.

double-precision floating point number. A 64-bit approximate representation of a real number.

downstream. The set of nodes in the syncpoint tree that is connected to the local DBMS as a participant in the execution of a two-phase commit.

| **DPSI.** Data-partitioned secondary index.

drain. The act of acquiring a locked resource by quiescing access to that object.

drain lock. A lock on a claim class that prevents a claim from occurring.

DRDA. Distributed Relational Database Architecture.

DRDA access. An open method of accessing distributed data that you can use to can connect to another database server to execute packages that were previously bound at the server location. You use the SQL CONNECT statement or an SQL statement with a three-part name to identify the server. Contrast with *private protocol access*.

DSN. (1) The default DB2 subsystem name. (2) The name of the TSO command processor of DB2. (3) The first three characters of DB2 module and macro names.

duration. A number that represents an interval of time. See also *date duration*, *labeled duration*, and *time duration*.

dynamic cursor. A named control structure that an application program uses to change the size of the result table and the order of its rows after the cursor is opened. Contrast with *static cursor.*

dynamic dump. A dump that is issued during the execution of a program, usually under the control of that program.

dynamic SQL. SQL statements that are prepared and executed within an application program while the program is executing. In dynamic SQL, the SQL source is contained in host language variables rather than being coded into the application program. The SQL statement can change several times during the application program's execution.

dynamic statement cache pool. A cache, located above the 2-GB storage line, that holds dynamic statements.

E

EA-enabled table space. A table space or index space that is enabled for extended addressability and that contains individual partitions (or pieces, for LOB table spaces) that are greater than 4 GB.

EB. See *exabyte*.

EBCDIC. Extended binary coded decimal interchange code. An encoding scheme that is used to represent character data in the z/OS, VM, VSE, and iSeries™ environments. Contrast with *ASCII* and *Unicode*.

e-business. The transformation of key business processes through the use of Internet technologies.

EDM pool. A pool of main storage that is used for database descriptors, application plans, authorization cache, application packages.

EID. Event identifier.

embedded SQL. SQL statements that are coded within an application program. See *static SQL*.

enclave. In Language Environment® , an independent collection of routines, one of which is designated as the main routine. An enclave is similar to a program or run unit.

encoding scheme. A set of rules to represent character data (ASCII, EBCDIC, or Unicode).

entity. A significant object of interest to an organization.

enumerated list. A set of DB2 objects that are defined with a LISTDEF utility control statement in which pattern-matching characters (*, %, _ or ?) are not used.

environment. A collection of names of logical and physical resources that are used to support the performance of a function.

environment handle. In DB2 ODBC, the data object that contains global information regarding the state of the application. An environment handle must be allocated before a connection handle can be allocated. Only one environment handle can be allocated per application.

EOM. End of memory.

EOT. End of task.

equijoin. A join operation in which the join-condition has the form *expression* = *expression*.

error page range. A range of pages that are considered to be physically damaged. DB2 does not allow users to access any pages that fall within this range.

escape character. The symbol that is used to enclose an SQL delimited identifier. The escape character is the double quotation mark ("), except in COBOL applications, where the user assigns the symbol, which is either a double quotation mark or an apostrophe (').

ESDS. Entry sequenced data set.

ESMT. External subsystem module table (in IMS).

EUR. IBM European Standards.

exabyte. For processor, real and virtual storage capacities and channel volume:
1 152 921 504 606 846 976 bytes or 2^{60}.

exception table. A table that holds rows that violate referential constraints or check constraints that the CHECK DATA utility finds.

exclusive lock. A lock that prevents concurrently executing application processes from reading or changing data. Contrast with *share lock*.

executable statement. An SQL statement that can be embedded in an application program, dynamically prepared and executed, or issued interactively.

execution context. In SQLJ, a Java object that can be used to control the execution of SQL statements.

exit routine. A user-written (or IBM-provided default) program that receives control from DB2 to perform specific functions. Exit routines run as extensions of DB2.

expanding conversion. A process that occurs when the length of a converted string is greater than that of the source string. For example, this process occurs when an ASCII mixed-data string that contains DBCS characters is converted to an EBCDIC mixed-data string; the converted string is longer because of the addition of shift codes.

explicit hierarchical locking. Locking that is used to make the parent-child relationship between resources known to IRLM. This kind of locking avoids global locking overhead when no inter-DB2 interest exists on a resource.

exposed name. A correlation name or a table or view name for which a correlation name is not specified. Names that are specified in a FROM clause are exposed or non-exposed.

expression. An operand or a collection of operators and operands that yields a single value.

extended recovery facility (XRF). A facility that minimizes the effect of failures in z/OS, VTAM , the host processor, or high-availability applications during sessions between high-availability applications and designated terminals. This facility provides an alternative subsystem to take over sessions from the failing subsystem.

Extensible Markup Language (XML). A standard metalanguage for defining markup languages that is a subset of Standardized General Markup Language (SGML). The less complex nature of XML makes it easier to write applications that handle document types, to author and manage structured information, and to transmit and share structured information across diverse computing environments.

external function. A function for which the body is written in a programming language that takes scalar argument values and produces a scalar result for each invocation. Contrast with *sourced function*, *built-in function*, and *SQL function*.

external procedure. A user-written application program that can be invoked with the SQL CALL statement, which is written in a programming language. Contrast with *SQL procedure*.

external routine. A user-defined function or stored procedure that is based on code that is written in an external programming language.

external subsystem module table (ESMT). In IMS, the table that specifies which attachment modules must be loaded.

F

failed member state. A state of a member of a data sharing group. When a member fails, the XCF permanently records the failed member state. This state usually means that the member's task, address space, or z/OS system terminated before the state changed from active to quiesced.

fallback. The process of returning to a previous release of DB2 after attempting or completing migration to a current release.

false global lock contention. A contention indication from the coupling facility when multiple lock names are hashed to the same indicator and when no real contention exists.

fan set. A direct physical access path to data, which is provided by an index, hash, or link; a fan set is the means by which the data manager supports the ordering of data.

federated database. The combination of a DB2 Universal Database server (in Linux, UNIX®, and Windows® environments) and multiple data sources to which the server sends queries. In a federated database system, a client application can use a single SQL statement to join data that is distributed across multiple database management systems and can view the data as if it were local.

fetch orientation. The specification of the desired placement of the cursor as part of a FETCH statement (for example, BEFORE, AFTER, NEXT, PRIOR, CURRENT, FIRST, LAST, ABSOLUTE, and RELATIVE).

field procedure. A user-written exit routine that is designed to receive a single value and transform (encode or decode) it in any way the user can specify.

filter factor. A number between zero and one that estimates the proportion of rows in a table for which a predicate is true.

fixed-length string. A character or graphic string whose length is specified and cannot be changed. Contrast with *varying-length string*.

FlashCopy. A function on the IBM Enterprise Storage Server® that can create a point-in-time copy of data while an application is running.

foreign key. A column or set of columns in a dependent table of a constraint relationship. The key must have the same number of columns, with the same descriptions, as the primary key of the parent table.

Each foreign key value must either match a parent key value in the related parent table or be null.

forest. An ordered set of subtrees of XML nodes.

forget. In a two-phase commit operation, (1) the vote that is sent to the prepare phase when the participant has not modified any data. The forget vote allows a participant to release locks and forget about the logical unit of work. This is also referred to as the read-only vote. (2) The response to the *committed* request in the second phase of the operation.

forward log recovery. The third phase of restart processing during which DB2 processes the log in a forward direction to apply all REDO log records.

free space. The total amount of unused space in a page; that is, the space that is not used to store records or control information is free space.

full outer join. The result of a join operation that includes the matched rows of both tables that are being joined and preserves the unmatched rows of both tables. See also *join*.

fullselect. A subselect, a values-clause, or a number of both that are combined by set operators. *Fullselect* specifies a result table. If UNION is not used, the result of the fullselect is the result of the specified subselect.

fully escaped mapping. A mapping from an SQL identifier to an XML name when the SQL identifier is a column name.

function. A mapping, which is embodied as a program (the function body) that is invocable by means of zero or more input values (arguments) to a single value (the result). See also *aggregate function* and *scalar function*.

Functions can be user-defined, built-in, or generated by DB2. (See also *built-in function*, *cast function*, *external function*, *sourced function*, *SQL function*, and *user-defined function*.)

function definer. The authorization ID of the owner of the schema of the function that is specified in the CREATE FUNCTION statement.

function implementer. The authorization ID of the owner of the function program and function package.

function package. A package that results from binding the DBRM for a function program.

function package owner. The authorization ID of the user who binds the function program's DBRM into a function package.

function resolution. The process, internal to the DBMS, by which a function invocation is bound to a particular function instance. This process uses the function name, the data types of the arguments, and a list of the applicable schema names (called the *SQL path*) to make the selection. This process is sometimes called *function selection*.

function selection. See *function resolution*.

function signature. The logical concatenation of a fully qualified function name with the data types of all of its parameters.

G

GB. Gigabyte (1 073 741 824 bytes).

GBP. Group buffer pool.

GBP-dependent. The status of a page set or page set partition that is dependent on the group buffer pool. Either read/write interest is active among DB2 subsystems for this page set, or the page set has changed pages in the group buffer pool that have not yet been cast out to disk.

generalized trace facility (GTF). A z/OS service program that records significant system events such as I/O interrupts, SVC interrupts, program interrupts, or external interrupts.

generic resource name. A name that VTAM uses to represent several application programs that provide the same function in order to handle session distribution and balancing in a Sysplex environment.

getpage. An operation in which DB2 accesses a data page.

global lock. A lock that provides concurrency control within and among DB2 subsystems. The scope of the lock is across all DB2 subsystems of a data sharing group.

global lock contention. Conflicts on locking requests between different DB2 members of a data sharing group when those members are trying to serialize shared resources.

governor. See *resource limit facility*.

graphic string. A sequence of DBCS characters.

gross lock. The *shared*, *update*, or *exclusive* mode locks on a table, partition, or table space.

group buffer pool (GBP). A coupling facility cache structure that is used by a data sharing group to cache data and to ensure that the data is consistent for all members.

group buffer pool duplexing. The ability to write data to two instances of a group buffer pool structure: a *primary group buffer pool* and a *secondary group buffer*

pool. z/OS publications refer to these instances as the "old" (for primary) and "new" (for secondary) structures.

group level. The release level of a data sharing group, which is established when the first member migrates to a new release.

group name. The z/OS XCF identifier for a data sharing group.

group restart. A restart of at least one member of a data sharing group after the loss of either locks or the shared communications area.

GTF. Generalized trace facility.

H

handle. In DB2 ODBC, a variable that refers to a data structure and associated resources. See also *statement handle*, *connection handle*, and *environment handle*.

help panel. A screen of information that presents tutorial text to assist a user at the workstation or terminal.

heuristic damage. The inconsistency in data between one or more participants that results when a heuristic decision to resolve an indoubt LUW at one or more participants differs from the decision that is recorded at the coordinator.

heuristic decision. A decision that forces indoubt resolution at a participant by means other than automatic resynchronization between coordinator and participant.

hole. A row of the result table that cannot be accessed because of a delete or an update that has been performed on the row. See also *delete hole* and *update hole*.

home address space. The area of storage that z/OS currently recognizes as *dispatched*.

host. The set of programs and resources that are available on a given TCP/IP instance.

host expression. A Java variable or expression that is referenced by SQL clauses in an SQLJ application program.

host identifier. A name that is declared in the host program.

host language. A programming language in which you can embed SQL statements.

host program. An application program that is written in a host language and that contains embedded SQL statements.

host structure. In an application program, a structure that is referenced by embedded SQL statements.

host variable. In an application program, an application variable that is referenced by embedded SQL statements.

host variable array. An array of elements, each of which corresponds to a value for a column. The dimension of the array determines the maximum number of rows for which the array can be used.

HSM. Hierarchical storage manager.

HTML. Hypertext Markup Language, a standard method for presenting Web data to users.

HTTP. Hypertext Transfer Protocol, a communication protocol that the Web uses.

I

ICF. Integrated catalog facility.

IDCAMS. An IBM program that is used to process access method services commands. It can be invoked as a job or jobstep, from a TSO terminal, or from within a user's application program.

IDCAMS LISTCAT. A facility for obtaining information that is contained in the access method services catalog.

identify. A request that an attachment service program in an address space that is separate from DB2 issues thorough the z/OS subsystem interface to inform DB2 of its existence and to initiate the process of becoming connected to DB2.

identity column. A column that provides a way for DB2 to automatically generate a numeric value for each row. The generated values are unique if cycling is not used. Identity columns are defined with the AS IDENTITY clause. Uniqueness of values can be ensured by defining a unique index that contains only the identity column. A table can have no more than one identity column.

IFCID. Instrumentation facility component identifier.

IFI. Instrumentation facility interface.

IFI call. An invocation of the instrumentation facility interface (IFI) by means of one of its defined functions.

IFP. IMS Fast Path.

image copy. An exact reproduction of all or part of a table space. DB2 provides utility programs to make full image copies (to copy the entire table space) or incremental image copies (to copy only those pages that have been modified since the last image copy).

implied forget. In the presumed-abort protocol, an implied response of *forget* to the second-phase *committed* request from the coordinator. The response is implied when the participant responds to any subsequent request from the coordinator.

IMS. Information Management System.

IMS attachment facility. A DB2 subcomponent that uses z/OS subsystem interface (SSI) protocols and cross-memory linkage to process requests from IMS to DB2 and to coordinate resource commitment.

IMS DB. Information Management System Database.

IMS TM. Information Management System Transaction Manager.

in-abort. A status of a unit of recovery. If DB2 fails after a unit of recovery begins to be rolled back, but before the process is completed, DB2 continues to back out the changes during restart.

in-commit. A status of a unit of recovery. If DB2 fails after beginning its phase 2 commit processing, it "knows," when restarted, that changes made to data are consistent. Such units of recovery are termed *in-commit*.

independent. An object (row, table, or table space) that is neither a parent nor a dependent of another object.

index. A set of pointers that are logically ordered by the values of a key. Indexes can provide faster access to data and can enforce uniqueness on the rows in a table.

index-controlled partitioning. A type of partitioning in which partition boundaries for a partitioned table are controlled by values that are specified on the CREATE INDEX statement. Partition limits are saved in the LIMITKEY column of the SYSIBM.SYSINDEXPART catalog table.

index key. The set of columns in a table that is used to determine the order of index entries.

index partition. A VSAM data set that is contained within a partitioning index space.

index space. A page set that is used to store the entries of one index.

indicator column. A 4-byte value that is stored in a base table in place of a LOB column.

indicator variable. A variable that is used to represent the null value in an application program. If the value for the selected column is null, a negative value is placed in the indicator variable.

indoubt. A status of a unit of recovery. If DB2 fails after it has finished its phase 1 commit processing and before it has started phase 2, only the commit coordinator knows if an individual unit of recovery is to be committed or rolled back. At emergency restart, if DB2 lacks the information it needs to make this decision, the status of the unit of recovery is *indoubt* until DB2 obtains this information from the coordinator. More than one unit of recovery can be indoubt at restart.

indoubt resolution. The process of resolving the status of an indoubt logical unit of work to either the committed or the rollback state.

inflight. A status of a unit of recovery. If DB2 fails before its unit of recovery completes phase 1 of the commit process, it merely backs out the updates of its unit of recovery at restart. These units of recovery are termed *inflight*.

inheritance. The passing downstream of class resources or attributes from a parent class in the class hierarchy to a child class.

initialization file. For DB2 ODBC applications, a file containing values that can be set to adjust the performance of the database manager.

inline copy. A copy that is produced by the LOAD or REORG utility. The data set that the inline copy produces is logically equivalent to a full image copy that is produced by running the COPY utility with read-only access (SHRLEVEL REFERENCE).

inner join. The result of a join operation that includes only the matched rows of both tables that are being joined. See also *join*.

inoperative package. A package that cannot be used because one or more user-defined functions or procedures that the package depends on were dropped. Such a package must be explicitly rebound. Contrast with *invalid package.*

insensitive cursor. A cursor that is not sensitive to inserts, updates, or deletes that are made to the underlying rows of a result table after the result table has been materialized.

insert trigger. A trigger that is defined with the triggering SQL operation INSERT.

install. The process of preparing a DB2 subsystem to operate as a z/OS subsystem.

installation verification scenario. A sequence of operations that exercises the main DB2 functions and tests whether DB2 was correctly installed.

instrumentation facility component identifier (IFCID). A value that names and identifies a trace record of an event that can be traced. As a parameter on the START TRACE and MODIFY TRACE commands, it specifies that the corresponding event is to be traced.

instrumentation facility interface (IFI). A programming interface that enables programs to obtain online trace data about DB2, to submit DB2 commands, and to pass data to DB2.

Interactive System Productivity Facility (ISPF). An IBM licensed program that provides interactive dialog services in a z/OS environment.

inter-DB2 R/W interest. A property of data in a table space, index, or partition that has been opened by more than one member of a data sharing group and that has been opened for writing by at least one of those members.

intermediate database server. The target of a request from a local application or a remote application requester that is forwarded to another database server. In the DB2 environment, the remote request is forwarded transparently to another database server if the object that is referenced by a three-part name does not reference the local location.

internationalization. The support for an encoding scheme that is able to represent the code points of characters from many different geographies and languages. To support all geographies, the Unicode standard requires more than 1 byte to represent a single character. See also *Unicode*.

internal resource lock manager (IRLM). A z/OS subsystem that DB2 uses to control communication and database locking.

International Organization for Standardization. An international body charged with creating standards to facilitate the exchange of goods and services as well as cooperation in intellectual, scientific, technological, and economic activity.

invalid package. A package that depends on an object (other than a user-defined function) that is dropped. Such a package is implicitly rebound on invocation. Contrast with *inoperative package.*

invariant character set. (1) A character set, such as the syntactic character set, whose code point assignments do not change from code page to code page. (2) A minimum set of characters that is available as part of all character sets.

IP address. A 4-byte value that uniquely identifies a TCP/IP host.

IRLM. Internal resource lock manager.

ISO. International Organization for Standardization.

isolation level. The degree to which a unit of work is isolated from the updating operations of other units of work. See also *cursor stability*, *read stability*, *repeatable read*, and *uncommitted read*.

ISPF. Interactive System Productivity Facility.

ISPF/PDF. Interactive System Productivity Facility/Program Development Facility.

iterator. In SQLJ, an object that contains the result set of a query. An iterator is equivalent to a cursor in other host languages.

iterator declaration clause. In SQLJ, a statement that generates an iterator declaration class. An iterator is an object of an iterator declaration class.

J

Japanese Industrial Standard. An encoding scheme that is used to process Japanese characters.

JAR. Java Archive.

Java Archive (JAR). A file format that is used for aggregating many files into a single file.

JCL. Job control language.

JDBC. A Sun Microsystems database application programming interface (API) for Java that allows programs to access database management systems by using callable SQL. JDBC does not require the use of an SQL preprocessor. In addition, JDBC provides an architecture that lets users add modules called *database drivers*, which link the application to their choice of database management systems at run time.

JES. Job Entry Subsystem.

JIS. Japanese Industrial Standard.

job control language (JCL). A control language that is used to identify a job to an operating system and to describe the job's requirements.

Job Entry Subsystem (JES). An IBM licensed program that receives jobs into the system and processes all output data that is produced by the jobs.

join. A relational operation that allows retrieval of data from two or more tables based on matching column values. See also *equijoin, full outer join, inner join, left outer join, outer join, and right outer join*.

K

KB. Kilobyte (1024 bytes).

Kerberos. A network authentication protocol that is designed to provide strong authentication for client/server applications by using secret-key cryptography.

Kerberos ticket. A transparent application mechanism that transmits the identity of an initiating principal to its target. A simple ticket contains the principal's

identity, a session key, a timestamp, and other information, which is sealed using the target's secret key.

key. A column or an ordered collection of columns that is identified in the description of a table, index, or referential constraint. The same column can be part of more than one key.

key-sequenced data set (KSDS). A VSAM file or data set whose records are loaded in key sequence and controlled by an index.

keyword. In SQL, a name that identifies an option that is used in an SQL statement.

KSDS. Key-sequenced data set.

L

labeled duration. A number that represents a duration of years, months, days, hours, minutes, seconds, or microseconds.

large object (LOB). A sequence of bytes representing bit data, single-byte characters, double-byte characters, or a mixture of single- and double-byte characters. A LOB can be up to 2 GB–1 byte in length. See also *BLOB*, *CLOB*, and *DBCLOB*.

last agent optimization. An optimized commit flow for either presumed-nothing or presumed-abort protocols in which the last agent, or final participant, becomes the commit coordinator. This flow saves at least one message.

latch. A DB2 internal mechanism for controlling concurrent events or the use of system resources.

LCID. Log control interval definition.

LDS. Linear data set.

leaf page. A page that contains pairs of keys and RIDs and that points to actual data. Contrast with *nonleaf page*.

left outer join. The result of a join operation that includes the matched rows of both tables that are being joined, and that preserves the unmatched rows of the first table. See also *join*.

limit key. The highest value of the index key for a partition.

linear data set (LDS). A VSAM data set that contains data but no control information. A linear data set can be accessed as a byte-addressable string in virtual storage.

linkage editor. A computer program for creating load modules from one or more object modules or load modules by resolving cross references among the modules and, if necessary, adjusting addresses.

link-edit. The action of creating a loadable computer program using a linkage editor.

list. A type of object, which DB2 utilities can process, that identifies multiple table spaces, multiple index spaces, or both. A list is defined with the LISTDEF utility control statement.

list structure. A coupling facility structure that lets data be shared and manipulated as elements of a queue.

LLE. Load list element.

L-lock. Logical lock.

load list element. A z/OS control block that controls the loading and deleting of a particular load module based on entry point names.

load module. A program unit that is suitable for loading into main storage for execution. The output of a linkage editor.

LOB. Large object.

LOB locator. A mechanism that allows an application program to manipulate a large object value in the database system. A LOB locator is a fullword integer value that represents a single LOB value. An application program retrieves a LOB locator into a host variable and can then apply SQL operations to the associated LOB value using the locator.

LOB lock. A lock on a LOB value.

LOB table space. A table space in an auxiliary table that contains all the data for a particular LOB column in the related base table.

local. A way of referring to any object that the local DB2 subsystem maintains. A *local table*, for example, is a table that is maintained by the local DB2 subsystem. Contrast with *remote*.

locale. The definition of a subset of a user's environment that combines a CCSID and characters that are defined for a specific language and country.

local lock. A lock that provides intra-DB2 concurrency control, but not inter-DB2 concurrency control; that is, its scope is a single DB2.

local subsystem. The unique relational DBMS to which the user or application program is directly connected (in the case of DB2, by one of the DB2 attachment facilities).

location. The unique name of a database server. An application uses the location name to access a DB2

database server. A database alias can be used to override the location name when accessing a remote server.

location alias. Another name by which a database server identifies itself in the network. Applications can use this name to access a DB2 database server.

lock. A means of controlling concurrent events or access to data. DB2 locking is performed by the IRLM.

lock duration. The interval over which a DB2 lock is held.

lock escalation. The promotion of a lock from a row, page, or LOB lock to a table space lock because the number of page locks that are concurrently held on a given resource exceeds a preset limit.

locking. The process by which the integrity of data is ensured. Locking prevents concurrent users from accessing inconsistent data.

lock mode. A representation for the type of access that concurrently running programs can have to a resource that a DB2 lock is holding.

lock object. The resource that is controlled by a DB2 lock.

lock promotion. The process of changing the size or mode of a DB2 lock to a higher, more restrictive level.

lock size. The amount of data that is controlled by a DB2 lock on table data; the value can be a row, a page, a LOB, a partition, a table, or a table space.

lock structure. A coupling facility data structure that is composed of a series of lock entries to support shared and exclusive locking for logical resources.

log. A collection of records that describe the events that occur during DB2 execution and that indicate their sequence. The information thus recorded is used for recovery in the event of a failure during DB2 execution.

log control interval definition. A suffix of the physical log record that tells how record segments are placed in the physical control interval.

logical claim. A claim on a logical partition of a nonpartitioning index.

logical data modeling. The process of documenting the comprehensive business information requirements in an accurate and consistent format. Data modeling is the first task of designing a database.

logical drain. A drain on a logical partition of a nonpartitioning index.

logical index partition. The set of all keys that reference the same data partition.

logical lock (L-lock). The lock type that transactions use to control intra- and inter-DB2 data concurrency between transactions. Contrast with *physical lock (P-lock)*.

logically complete. A state in which the concurrent copy process is finished with the initialization of the target objects that are being copied. The target objects are available for update.

logical page list (LPL). A list of pages that are in error and that cannot be referenced by applications until the pages are recovered. The page is in *logical error* because the actual media (coupling facility or disk) might not contain any errors. Usually a connection to the media has been lost.

logical partition. A set of key or RID pairs in a nonpartitioning index that are associated with a particular partition.

logical recovery pending (LRECP). The state in which the data and the index keys that reference the data are inconsistent.

logical unit (LU). An access point through which an application program accesses the SNA network in order to communicate with another application program.

logical unit of work (LUW). The processing that a program performs between synchronization points.

logical unit of work identifier (LUWID). A name that uniquely identifies a thread within a network. This name consists of a fully-qualified LU network name, an LUW instance number, and an LUW sequence number.

log initialization. The first phase of restart processing during which DB2 attempts to locate the current end of the log.

log record header (LRH). A prefix, in every logical record, that contains control information.

log record sequence number (LRSN). A unique identifier for a log record that is associated with a data sharing member. DB2 uses the LRSN for recovery in the data sharing environment.

log truncation. A process by which an explicit starting RBA is established. This RBA is the point at which the next byte of log data is to be written.

LPL. Logical page list.

LRECP. Logical recovery pending.

LRH. Log record header.

LRSN. Log record sequence number.

LU. Logical unit.

LU name. Logical unit name, which is the name by which VTAM refers to a node in a network. Contrast with *location name*.

LUW. Logical unit of work.

LUWID. Logical unit of work identifier.

M

mapping table. A table that the REORG utility uses to map the associations of the RIDs of data records in the original copy and in the shadow copy. This table is created by the user.

mass delete. The deletion of all rows of a table.

master terminal. The IMS logical terminal that has complete control of IMS resources during online operations.

master terminal operator (MTO). See *master terminal.*

materialize. (1) The process of putting rows from a view or nested table expression into a work file for additional processing by a query.

(2) The placement of a LOB value into contiguous storage. Because LOB values can be very large, DB2 avoids materializing LOB data until doing so becomes absolutely necessary.

materialized query table. A table that is used to contain information that is derived and can be summarized from one or more source tables.

MB. Megabyte (1 048 576 bytes).

MBCS. Multibyte character set. UTF-8 is an example of an MBCS. Characters in UTF-8 can range from 1 to 4 bytes in DB2.

member name. The z/OS XCF identifier for a particular DB2 subsystem in a data sharing group.

menu. A displayed list of available functions for selection by the operator. A menu is sometimes called a *menu panel*.

metalanguage. A language that is used to create other specialized languages.

migration. The process of converting a subsystem with a previous release of DB2 to an updated or current release. In this process, you can acquire the functions of the updated or current release without losing the data that you created on the previous release.

mixed data string. A character string that can contain both single-byte and double-byte characters.

MLPA. Modified link pack area.

MODEENT. A VTAM macro instruction that associates a logon mode name with a set of parameters representing session protocols. A set of MODEENT macro instructions defines a logon mode table.

modeling database. A DB2 database that you create on your workstation that you use to model a DB2 UDB for z/OS subsystem, which can then be evaluated by the Index Advisor.

mode name. A VTAM name for the collection of physical and logical characteristics and attributes of a session.

modify locks. An L-lock or P-lock with a MODIFY attribute. A list of these active locks is kept at all times in the coupling facility lock structure. If the requesting DB2 subsystem fails, that DB2 subsystem's modify locks are converted to retained locks.

MPP. Message processing program (in IMS).

MTO. Master terminal operator.

multibyte character set (MBCS). A character set that represents single characters with more than a single byte. Contrast with *single-byte character set* and *double-byte character set*. See also *Unicode*.

multidimensional analysis. The process of assessing and evaluating an enterprise on more than one level.

Multiple Virtual Storage. An element of the z/OS operating system. This element is also called the Base Control Program (BCP).

multisite update. Distributed relational database processing in which data is updated in more than one location within a single unit of work.

multithreading. Multiple TCBs that are executing one copy of DB2 ODBC code concurrently (sharing a processor) or in parallel (on separate central processors).

must-complete. A state during DB2 processing in which the entire operation must be completed to maintain data integrity.

mutex. Pthread mutual exclusion; a lock. A Pthread mutex variable is used as a locking mechanism to allow serialization of critical sections of code by temporarily blocking the execution of all but one thread.

MVS. See *Multiple Virtual Storage*.

N

negotiable lock. A lock whose mode can be downgraded, by agreement among contending users, to be compatible to all. A physical lock is an example of a negotiable lock.

nested table expression. A fullselect in a FROM clause (surrounded by parentheses).

network identifier (NID). The network ID that is assigned by IMS or CICS, or if the connection type is RRSAF, the RRS unit of recovery ID (URID).

NID. Network identifier.

nonleaf page. A page that contains keys and page numbers of other pages in the index (either leaf or nonleaf pages). Nonleaf pages never point to actual data.

nonpartitioned index. An index that is not physically partitioned. Both partitioning indexes and secondary indexes can be nonpartitioned.

nonscrollable cursor. A cursor that can be moved only in a forward direction. Nonscrollable cursors are sometimes called forward-only cursors or serial cursors.

normalization. A key step in the task of building a logical relational database design. Normalization helps you avoid redundancies and inconsistencies in your data. An entity is normalized if it meets a set of constraints for a particular normal form (first normal form, second normal form, and so on). Contrast with *denormalization*.

nondeterministic function. A user-defined function whose result is not solely dependent on the values of the input arguments. That is, successive invocations with the same argument values can produce a different answer. this type of function is sometimes called a *variant* function. Contrast this with a *deterministic function* (sometimes called a *not-variant function*), which always produces the same result for the same inputs.

not-variant function. See *deterministic function*.

NPSI. See *nonpartitioned secondary index.*

NRE. Network recovery element.

NUL. The null character ('\0'), which is represented by the value X'00'. In C, this character denotes the end of a string.

null. A special value that indicates the absence of information.

NULLIF. A scalar function that evaluates two passed expressions, returning either NULL if the arguments are equal or the value of the first argument if they are not.

null-terminated host variable. A varying-length host variable in which the end of the data is indicated by a null terminator.

null terminator. In C, the value that indicates the end of a string. For EBCDIC, ASCII, and Unicode UTF-8 strings, the null terminator is a single-byte value (X'00'). For Unicode UCS-2 (wide) strings, the null terminator is a double-byte value (X'0000').

O

OASN (origin application schedule number). In IMS, a 4-byte number that is assigned sequentially to each IMS schedule since the last cold start of IMS. The OASN is used as an identifier for a unit of work. In an 8-byte format, the first 4 bytes contain the schedule number and the last 4 bytes contain the number of IMS sync points (*commit points*) during the current schedule. The OASN is part of the NID for an IMS connection.

ODBC. Open Database Connectivity.

ODBC driver. A dynamically-linked library (DLL) that implements ODBC function calls and interacts with a data source.

OBID. Data object identifier.

Open Database Connectivity (ODBC). A Microsoft® database application programming interface (API) for C that allows access to database management systems by using callable SQL. ODBC does not require the use of an SQL preprocessor. In addition, ODBC provides an architecture that lets users add modules called *database drivers*, which link the application to their choice of database management systems at run time. This means that applications no longer need to be directly linked to the modules of all the database management systems that are supported.

ordinary identifier. An uppercase letter followed by zero or more characters, each of which is an uppercase letter, a digit, or the underscore character. An ordinary identifier must not be a reserved word.

ordinary token. A numeric constant, an ordinary identifier, a host identifier, or a keyword.

originating task. In a parallel group, the primary agent that receives data from other execution units (referred to as *parallel tasks*) that are executing portions of the query in parallel.

OS/390. Operating System/390®.

outer join. The result of a join operation that includes the matched rows of both tables that are being joined and preserves some or all of the unmatched rows of the tables that are being joined. See also *join*.

overloaded function. A function name for which multiple function instances exist.

P

package. An object containing a set of SQL statements that have been statically bound and that is available for processing. A package is sometimes also called an *application package*.

package list. An ordered list of package names that may be used to extend an application plan.

package name. The name of an object that is created by a BIND PACKAGE or REBIND PACKAGE command. The object is a bound version of a database request module (DBRM). The name consists of a location name, a collection ID, a package ID, and a version ID.

page. A unit of storage within a table space (4 KB, 8 KB, 16 KB, or 32 KB) or index space (4 KB). In a table space, a page contains one or more rows of a table. In a LOB table space, a LOB value can span more than one page, but no more than one LOB value is stored on a page.

page set. Another way to refer to a table space or index space. Each page set consists of a collection of VSAM data sets.

page set recovery pending (PSRCP). A restrictive state of an index space. In this case, the entire page set must be recovered. Recovery of a logical part is prohibited.

panel. A predefined display image that defines the locations and characteristics of display fields on a display surface (for example, a *menu panel*).

parallel complex. A cluster of machines that work together to handle multiple transactions and applications.

parallel group. A set of consecutive operations that execute in parallel and that have the same number of parallel tasks.

parallel I/O processing. A form of I/O processing in which DB2 initiates multiple concurrent requests for a single user query and performs I/O processing concurrently (in *parallel*) on multiple data partitions.

parallelism assistant. In Sysplex query parallelism, a DB2 subsystem that helps to process parts of a parallel query that originates on another DB2 subsystem in the data sharing group.

parallelism coordinator. In Sysplex query parallelism, the DB2 subsystem from which the parallel query originates.

Parallel Sysplex. A set of z/OS systems that communicate and cooperate with each other through certain multisystem hardware components and software services to process customer workloads.

parallel task. The execution unit that is dynamically created to process a query in parallel. A parallel task is implemented by a z/OS service request block.

parameter marker. A question mark (?) that appears in a statement string of a dynamic SQL statement. The question mark can appear where a host variable could appear if the statement string were a static SQL statement.

parameter-name. An SQL identifier that designates a parameter in an SQL procedure or an SQL function.

parent key. A primary key or unique key in the parent table of a referential constraint. The values of a parent key determine the valid values of the foreign key in the referential constraint.

parent lock. For explicit hierarchical locking, a lock that is held on a resource that might have child locks that are lower in the hierarchy. A parent lock is usually the table space lock or the partition intent lock. See also *child lock*.

parent row. A row whose primary key value is the foreign key value of a dependent row.

parent table. A table whose primary key is referenced by the foreign key of a dependent table.

parent table space. A table space that contains a parent table. A table space containing a dependent of that table is a dependent table space.

participant. An entity other than the commit coordinator that takes part in the commit process. The term participant is synonymous with *agent* in SNA.

partition. A portion of a page set. Each partition corresponds to a single, independently extendable data set. Partitions can be extended to a maximum size of 1, 2, or 4 GB, depending on the number of partitions in the partitioned page set. All partitions of a given page set have the same maximum size.

partitioned data set (PDS). A data set in disk storage that is divided into partitions, which are called members. Each partition can contain a program, part of a program, or data. The term partitioned data set is synonymous with program library.

partitioned index. An index that is physically partitioned. Both partitioning indexes and secondary indexes can be partitioned.

partitioned page set. A partitioned table space or an index space. Header pages, space map pages, data pages, and index pages reference data only within the scope of the partition.

partitioned table space. A table space that is subdivided into parts (based on index key range), each of which can be processed independently by utilities.

partitioning index. An index in which the leftmost columns are the partitioning columns of the table. The index can be partitioned or nonpartitioned.

partition pruning. The removal from consideration of inapplicable partitions through setting up predicates in a query on a partitioned table to access only certain partitions to satisfy the query.

partner logical unit. An access point in the SNA network that is connected to the local DB2 subsystem by way of a VTAM conversation.

path. See *SQL path*.

PCT. Program control table (in CICS).

PDS. Partitioned data set.

piece. A data set of a nonpartitioned page set.

physical claim. A claim on an entire nonpartitioning index.

physical consistency. The state of a page that is not in a partially changed state.

physical drain. A drain on an entire nonpartitioning index.

physical lock (P-lock). A type of lock that DB2 acquires to provide consistency of data that is cached in different DB2 subsystems. Physical locks are used only in data sharing environments. Contrast with *logical lock (L-lock)*.

physical lock contention. Conflicting states of the requesters for a physical lock. See also *negotiable lock*.

physically complete. The state in which the concurrent copy process is completed and the output data set has been created.

plan. See *application plan*.

plan allocation. The process of allocating DB2 resources to a plan in preparation for execution.

plan member. The bound copy of a DBRM that is identified in the member clause.

plan name. The name of an application plan.

plan segmentation. The dividing of each plan into sections. When a section is needed, it is independently brought into the EDM pool.

P-lock. Physical lock.

PLT. Program list table (in CICS).

point of consistency. A time when all recoverable data that an application accesses is consistent with other data. The term point of consistency is synonymous with *sync point* or *commit point*.

policy. See *CFRM policy*.

Portable Operating System Interface (POSIX). The IEEE operating system interface standard, which defines the Pthread standard of threading. See also *Pthread*.

POSIX. Portable Operating System Interface.

postponed abort UR. A unit of recovery that was inflight or in-abort, was interrupted by system failure or cancellation, and did not complete backout during restart.

PPT. (1) Processing program table (in CICS). (2) Program properties table (in z/OS).

precision. In SQL, the total number of digits in a decimal number (called the *size* in the C language). In the C language, the number of digits to the right of the decimal point (called the *scale* in SQL). The DB2 library uses the SQL terms.

precompilation. A processing of application programs containing SQL statements that takes place before compilation. SQL statements are replaced with statements that are recognized by the host language compiler. Output from this precompilation includes source code that can be submitted to the compiler and the database request module (DBRM) that is input to the bind process.

predicate. An element of a search condition that expresses or implies a comparison operation.

prefix. A code at the beginning of a message or record.

preformat. The process of preparing a VSAM ESDS for DB2 use, by writing specific data patterns.

prepare. The first phase of a two-phase commit process in which all participants are requested to prepare for commit.

prepared SQL statement. A named object that is the executable form of an SQL statement that has been processed by the PREPARE statement.

presumed-abort. An optimization of the presumed-nothing two-phase commit protocol that reduces the number of recovery log records, the duration of state maintenance, and the number of messages between coordinator and participant. The optimization also modifies the indoubt resolution responsibility.

presumed-nothing. The standard two-phase commit protocol that defines coordinator and participant responsibilities, relative to logical unit of work states, recovery logging, and indoubt resolution.

primary authorization ID. The authorization ID that is used to identify the application process to DB2.

primary group buffer pool. For a duplexed group buffer pool, the structure that is used to maintain the coherency of cached data. This structure is used for page registration and cross-invalidation. The z/OS equivalent is *old* structure. Compare with *secondary group buffer pool*.

primary index. An index that enforces the uniqueness of a primary key.

primary key. In a relational database, a unique, nonnull key that is part of the definition of a table. A table cannot be defined as a parent unless it has a unique key or primary key.

principal. An entity that can communicate securely with another entity. In Kerberos, principals are represented as entries in the Kerberos registry database and include users, servers, computers, and others.

principal name. The name by which a principal is known to the DCE security services.

private connection. A communications connection that is specific to DB2.

private protocol access. A method of accessing distributed data by which you can direct a query to another DB2 system. Contrast with *DRDA access*.

private protocol connection. A DB2 private connection of the application process. See also *private connection*.

privilege. The capability of performing a specific function, sometimes on a specific object. The types of privileges are:

explicit privileges, which have names and are held as the result of SQL GRANT and REVOKE statements. For example, the SELECT privilege.

implicit privileges, which accompany the ownership of an object, such as the privilege to drop a synonym that one owns, or the holding of an authority, such as the privilege of SYSADM authority to terminate any utility job.

privilege set. For the installation SYSADM ID, the set of all possible privileges. For any other authorization ID, the set of all privileges that are recorded for that ID in the DB2 catalog.

process. In DB2, the unit to which DB2 allocates resources and locks. Sometimes called an *application process*, a process involves the execution of one or more programs. The execution of an SQL statement is always associated with some process. The means of initiating and terminating a process are dependent on the environment.

program. A single, compilable collection of executable statements in a programming language.

program temporary fix (PTF). A solution or bypass of a problem that is diagnosed as a result of a defect in a current unaltered release of a licensed program. An authorized program analysis report (APAR) fix is corrective service for an existing problem. A PTF is preventive service for problems that might be encountered by other users of the product. A PTF is *temporary*, because a permanent fix is usually not incorporated into the product until its next release.

protected conversation. A VTAM conversation that supports two-phase commit flows.

PSRCP. Page set recovery pending.

PTF. Program temporary fix.

Pthread. The POSIX threading standard model for splitting an application into subtasks. The Pthread standard includes functions for creating threads, terminating threads, synchronizing threads through locking, and other thread control facilities.

Q

QMF™. Query Management Facility.

QSAM. Queued sequential access method.

query. A component of certain SQL statements that specifies a result table.

query block. The part of a query that is represented by one of the FROM clauses. Each FROM clause can have multiple query blocks, depending on DB2's internal processing of the query.

query CP parallelism. Parallel execution of a single query, which is accomplished by using multiple tasks. See also *Sysplex query parallelism*.

query I/O parallelism. Parallel access of data, which is accomplished by triggering multiple I/O requests within a single query.

queued sequential access method (QSAM). An extended version of the basic sequential access method (BSAM). When this method is used, a queue of data blocks is formed. Input data blocks await processing, and output data blocks await transfer to auxiliary storage or to an output device.

quiesce point. A point at which data is consistent as a result of running the DB2 QUIESCE utility.

quiesced member state. A state of a member of a data sharing group. An active member becomes quiesced when a STOP DB2 command takes effect without a failure. If the member's task, address space, or z/OS system fails before the command takes effect, the member state is failed.

R

RACF. Resource Access Control Facility, which is a component of the z/OS Security Server.

RAMAC®. IBM family of enterprise disk storage system products.

RBA. Relative byte address.

RCT. Resource control table (in CICS attachment facility).

RDB. Relational database.

RDBMS. Relational database management system.

RDBNAM. Relational database name.

RDF. Record definition field.

read stability (RS). An isolation level that is similar to repeatable read but does not completely isolate an application process from all other concurrently executing application processes. Under level RS, an application that issues the same query more than once might read additional rows that were inserted and committed by a concurrently executing application process.

rebind. The creation of a new application plan for an application program that has been bound previously. If, for example, you have added an index for a table that your application accesses, you must rebind the application in order to take advantage of that index.

rebuild. The process of reallocating a coupling facility structure. For the shared communications area (SCA) and lock structure, the structure is repopulated; for the group buffer pool, changed pages are usually cast out to disk, and the new structure is populated only with changed pages that were not successfully cast out.

RECFM. Record format.

record. The storage representation of a row or other data.

record identifier (RID). A unique identifier that DB2 uses internally to identify a row of data in a table. Compare with *row ID*.

record identifier (RID) pool. An area of main storage that is used for sorting record identifiers during list-prefetch processing.

record length. The sum of the length of all the columns in a table, which is the length of the data as it is physically stored in the database. Records can be fixed length or varying length, depending on how the columns are defined. If all columns are fixed-length columns, the record is a fixed-length record. If one or more columns are varying-length columns, the record is a varying-length column.

Recoverable Resource Manager Services attachment facility (RRSAF). A DB2 subcomponent that uses Resource Recovery Services to coordinate resource commitment between DB2 and all other resource managers that also use RRS in a z/OS system.

recovery. The process of rebuilding databases after a system failure.

recovery log. A collection of records that describes the events that occur during DB2 execution and indicates their sequence. The recorded information is used for recovery in the event of a failure during DB2 execution.

recovery manager. (1) A subcomponent that supplies coordination services that control the interaction of DB2 resource managers during commit, abort, checkpoint, and restart processes. The recovery manager also supports the recovery mechanisms of other subsystems (for example, IMS) by acting as a participant in the other subsystem's process for protecting data that has reached a point of consistency. (2) A coordinator or a participant (or both), in the execution of a two-phase commit, that can access a recovery log that maintains the state of the logical unit of work and names the immediate upstream coordinator and downstream participants.

recovery pending (RECP). A condition that prevents SQL access to a table space that needs to be recovered.

recovery token. An identifier for an element that is used in recovery (for example, NID or URID).

RECP. Recovery pending.

redo. A state of a unit of recovery that indicates that changes are to be reapplied to the disk media to ensure data integrity.

reentrant. Executable code that can reside in storage as one shared copy for all threads. Reentrant code is not self-modifying and provides separate storage areas for each thread. Reentrancy is a compiler and operating system concept, and reentrancy alone is not enough to guarantee logically consistent results when multithreading. See also *threadsafe*.

referential constraint. The requirement that nonnull values of a designated foreign key are valid only if they equal values of the primary key of a designated table.

referential integrity. The state of a database in which all values of all foreign keys are valid. Maintaining referential integrity requires the enforcement of referential constraints on all operations that change the data in a table on which the referential constraints are defined.

referential structure. A set of tables and relationships that includes at least one table and, for every table in the set, all the relationships in which that table participates and all the tables to which it is related.

refresh age. The time duration between the current time and the time during which a materialized query table was last refreshed.

registry. See *registry database*.

registry database. A database of security information about principals, groups, organizations, accounts, and security policies.

relational database (RDB). A database that can be perceived as a set of tables and manipulated in accordance with the relational model of data.

relational database management system (RDBMS). A collection of hardware and software that organizes and provides access to a relational database.

relational database name (RDBNAM). A unique identifier for an RDBMS within a network. In DB2, this must be the value in the LOCATION column of table SYSIBM.LOCATIONS in the CDB. DB2 publications refer to the name of another RDBMS as a LOCATION value or a location name.

relationship. A defined connection between the rows of a table or the rows of two tables. A relationship is the internal representation of a referential constraint.

relative byte address (RBA). The offset of a data record or control interval from the beginning of the storage space that is allocated to the data set or file to which it belongs.

remigration. The process of returning to a current release of DB2 following a fallback to a previous release. This procedure constitutes another migration process.

remote. Any object that is maintained by a remote DB2 subsystem (that is, by a DB2 subsystem other than the local one). A *remote view*, for example, is a view that is maintained by a remote DB2 subsystem. Contrast with *local*.

remote attach request. A request by a remote location to attach to the local DB2 subsystem. Specifically, the request that is sent is an SNA Function Management Header 5.

remote subsystem. Any relational DBMS, except the *local subsystem*, with which the user or application can communicate. The subsystem need not be remote in any physical sense, and might even operate on the same processor under the same z/OS system.

reoptimization. The DB2 process of reconsidering the access path of an SQL statement at run time; during reoptimization, DB2 uses the values of host variables, parameter markers, or special registers.

REORG pending (REORP). A condition that restricts SQL access and most utility access to an object that must be reorganized.

REORP. REORG pending.

repeatable read (RR). The isolation level that provides maximum protection from other executing application programs. When an application program executes with repeatable read protection, rows that the program references cannot be changed by other programs until the program reaches a commit point.

repeating group. A situation in which an entity includes multiple attributes that are inherently the same. The presence of a repeating group violates the requirement of first normal form. In an entity that satisfies the requirement of first normal form, each attribute is independent and unique in its meaning and its name. See also *normalization*.

replay detection mechanism. A method that allows a principal to detect whether a request is a valid request from a source that can be trusted or whether an untrustworthy entity has captured information from a previous exchange and is replaying the information exchange to gain access to the principal.

request commit. The vote that is submitted to the prepare phase if the participant has modified data and is prepared to commit or roll back.

requester. The source of a request to access data at a remote server. In the DB2 environment, the requester function is provided by the distributed data facility.

resource. The object of a lock or claim, which could be a table space, an index space, a data partition, an index partition, or a logical partition.

resource allocation. The part of plan allocation that deals specifically with the database resources.

resource control table (RCT). A construct of the CICS attachment facility, created by site-provided macro parameters, that defines authorization and access attributes for transactions or transaction groups.

resource definition online. A CICS feature that you use to define CICS resources online without assembling tables.

resource limit facility (RLF). A portion of DB2 code that prevents dynamic manipulative SQL statements from exceeding specified time limits. The resource limit facility is sometimes called the governor.

resource limit specification table (RLST). A site-defined table that specifies the limits to be enforced by the resource limit facility.

resource manager. (1) A function that is responsible for managing a particular resource and that guarantees the consistency of all updates made to recoverable resources within a logical unit of work. The resource that is being managed can be physical (for example, disk or main storage) or logical (for example, a particular type of system service). (2) A participant, in the execution of a two-phase commit, that has recoverable resources that could have been modified. The resource manager has access to a recovery log so that it can commit or roll back the effects of the logical unit of work to the recoverable resources.

restart pending (RESTP). A restrictive state of a page set or partition that indicates that restart (backout) work needs to be performed on the object. All access to the page set or partition is denied except for access by the:
- RECOVER POSTPONED command
- Automatic online backout (which DB2 invokes after restart if the system parameter LBACKOUT=AUTO)

RESTP. Restart pending.

result set. The set of rows that a stored procedure returns to a client application.

result set locator. A 4-byte value that DB2 uses to uniquely identify a query result set that a stored procedure returns.

result table. The set of rows that are specified by a SELECT statement.

retained lock. A MODIFY lock that a DB2 subsystem was holding at the time of a subsystem failure. The lock is retained in the coupling facility lock structure across a DB2 failure.

RID. Record identifier.

RID pool. Record identifier pool.

right outer join. The result of a join operation that includes the matched rows of both tables that are being joined and preserves the unmatched rows of the second join operand. See also *join*.

RLF. Resource limit facility.

RLST. Resource limit specification table.

RMID. Resource manager identifier.

RO. Read-only access.

rollback. The process of restoring data that was changed by SQL statements to the state at its last commit point. All locks are freed. Contrast with *commit*.

root page. The index page that is at the highest level (or the beginning point) in an index.

routine. A term that refers to either a user-defined function or a stored procedure.

row. The horizontal component of a table. A row consists of a sequence of values, one for each column of the table.

ROWID. Row identifier.

row identifier (ROWID). A value that uniquely identifies a row. This value is stored with the row and never changes.

row lock. A lock on a single row of data.

| **rowset.** A set of rows for which a cursor position is
| established.

| **rowset cursor.** A cursor that is defined so that one or
| more rows can be returned as a rowset for a single
| FETCH statement, and the cursor is positioned on the
| set of rows that is fetched.

| **rowset-positioned access.** The ability to retrieve
| multiple rows from a single FETCH statement.

| **row-positioned access.** The ability to retrieve a single
| row from a single FETCH statement.

row trigger. A trigger that is defined with the trigger granularity FOR EACH ROW.

RRE. Residual recovery entry (in IMS).

RRSAF. Recoverable Resource Manager Services attachment facility.

RS. Read stability.

RTT. Resource translation table.

RURE. Restart URE.

S

savepoint. A named entity that represents the state of data and schemas at a particular point in time within a unit of work. SQL statements exist to set a savepoint, release a savepoint, and restore data and schemas to the state that the savepoint represents. The restoration of data and schemas to a savepoint is usually referred to as *rolling back to a savepoint*.

SBCS. Single-byte character set.

SCA. Shared communications area.

scalar function. An SQL operation that produces a
single value from another value and is expressed as a
function name, followed by a list of arguments that are
enclosed in parentheses. Contrast with *aggregate*
function.

scale. In SQL, the number of digits to the right of the decimal point (called the *precision* in the C language). The DB2 library uses the SQL definition.

schema. (1) The organization or structure of a database. (2) A logical grouping for user-defined functions, distinct types, triggers, and stored procedures. When an object of one of these types is created, it is assigned to one schema, which is determined by the name of the object. For example, the following statement creates a distinct type T in schema C:

```
CREATE DISTINCT TYPE C.T ...
```

scrollability. The ability to use a cursor to fetch in either a forward or backward direction. The FETCH statement supports multiple fetch orientations to indicate the new position of the cursor. See also *fetch orientation*.

scrollable cursor. A cursor that can be moved in both a forward and a backward direction.

SDWA. System diagnostic work area.

search condition. A criterion for selecting rows from a table. A search condition consists of one or more predicates.

secondary authorization ID. An authorization ID that has been associated with a primary authorization ID by an authorization exit routine.

secondary group buffer pool. For a duplexed group buffer pool, the structure that is used to back up changed pages that are written to the primary group buffer pool. No page registration or cross-invalidation occurs using the secondary group buffer pool. The z/OS equivalent is *new* structure.

secondary index. A nonpartitioning index on a partitioned table.

section. The segment of a plan or package that contains the executable structures for a single SQL statement. For most SQL statements, one section in the plan exists for each SQL statement in the source program. However, for cursor-related statements, the DECLARE, OPEN, FETCH, and CLOSE statements reference the same section because they each refer to the SELECT statement that is named in the DECLARE CURSOR statement. SQL statements such as COMMIT, ROLLBACK, and some SET statements do not use a section.

segment. A group of pages that holds rows of a single table. See also *segmented table space*.

segmented table space. A table space that is divided into equal-sized groups of pages called segments. Segments are assigned to tables so that rows of different tables are never stored in the same segment.

self-referencing constraint. A referential constraint that defines a relationship in which a table is a dependent of itself.

self-referencing table. A table with a self-referencing constraint.

sensitive cursor. A cursor that is sensitive to changes that are made to the database after the result table has been materialized.

sequence. A user-defined object that generates a sequence of numeric values according to user specifications.

sequential data set. A non-DB2 data set whose records are organized on the basis of their successive physical positions, such as on magnetic tape. Several of the DB2 database utilities require sequential data sets.

sequential prefetch. A mechanism that triggers consecutive asynchronous I/O operations. Pages are fetched before they are required, and several pages are read with a single I/O operation.

serial cursor. A cursor that can be moved only in a forward direction.

serialized profile. A Java object that contains SQL statements and descriptions of host variables. The SQLJ translator produces a serialized profile for each connection context.

server. The target of a request from a remote requester. In the DB2 environment, the server function is provided by the distributed data facility, which is used to access DB2 data from remote applications.

server-side programming. A method for adding DB2 data into dynamic Web pages.

service class. An eight-character identifier that is used by the z/OS Workload Manager to associate user performance goals with a particular DDF thread or stored procedure. A service class is also used to classify work on parallelism assistants.

service request block. A unit of work that is scheduled to execute in another address space.

session. A link between two nodes in a VTAM network.

session protocols. The available set of SNA communication requests and responses.

shared communications area (SCA). A coupling facility list structure that a DB2 data sharing group uses for inter-DB2 communication.

share lock. A lock that prevents concurrently executing application processes from changing data, but not from reading data. Contrast with *exclusive lock*.

shift-in character. A special control character (X'0F') that is used in EBCDIC systems to denote that the subsequent bytes represent SBCS characters. See also *shift-out character*.

shift-out character. A special control character (X'0E') that is used in EBCDIC systems to denote that the subsequent bytes, up to the next shift-in control character, represent DBCS characters. See also *shift-in character*.

sign-on. A request that is made on behalf of an individual CICS or IMS application process by an attachment facility to enable DB2 to verify that it is authorized to use DB2 resources.

simple page set. A nonpartitioned page set. A simple page set initially consists of a single data set (page set piece). If and when that data set is extended to 2 GB, another data set is created, and so on, up to a total of 32 data sets. DB2 considers the data sets to be a single contiguous linear address space containing a maximum of 64 GB. Data is stored in the next available location within this address space without regard to any partitioning scheme.

simple table space. A table space that is neither partitioned nor segmented.

single-byte character set (SBCS). A set of characters in which each character is represented by a single byte. Contrast with *double-byte character set* or *multibyte character set*.

single-precision floating point number. A 32-bit approximate representation of a real number.

size. In the C language, the total number of digits in a decimal number (called the *precision* in SQL). The DB2 library uses the SQL term.

SMF. System Management Facilities.

SMP/E. System Modification Program/Extended.

SMS. Storage Management Subsystem.

SNA. Systems Network Architecture.

SNA network. The part of a network that conforms to the formats and protocols of Systems Network Architecture (SNA).

socket. A callable TCP/IP programming interface that TCP/IP network applications use to communicate with remote TCP/IP partners.

sourced function. A function that is implemented by another built-in or user-defined function that is already known to the database manager. This function can be a scalar function or a column (aggregating) function; it returns a single value from a set of values (for example, MAX or AVG). Contrast with *built-in function, external function,* and *SQL function*.

source program. A set of host language statements and SQL statements that is processed by an SQL precompiler.

source table. A table that can be a base table, a view, a table expression, or a user-defined table function.

source type. An existing type that DB2 uses to internally represent a distinct type.

space. A sequence of one or more blank characters.

special register. A storage area that DB2 defines for an application process to use for storing information that can be referenced in SQL statements. Examples of special registers are USER and CURRENT DATE.

specific function name. A particular user-defined function that is known to the database manager by its specific name. Many specific user-defined functions can have the same function name. When a user-defined function is defined to the database, every function is assigned a specific name that is unique within its schema. Either the user can provide this name, or a default name is used.

SPUFI. SQL Processor Using File Input.

SQL. Structured Query Language.

SQL authorization ID (SQL ID). The authorization ID that is used for checking dynamic SQL statements in some situations.

SQLCA. SQL communication area.

SQL communication area (SQLCA). A structure that is used to provide an application program with information about the execution of its SQL statements.

SQL connection. An association between an application process and a local or remote application server or database server.

SQLDA. SQL descriptor area.

SQL descriptor area (SQLDA). A structure that describes input variables, output variables, or the columns of a result table.

SQL escape character. The symbol that is used to enclose an SQL delimited identifier. This symbol is the double quotation mark ("). See also *escape character*.

SQL function. A user-defined function in which the CREATE FUNCTION statement contains the source code. The source code is a single SQL expression that evaluates to a single value. The SQL user-defined function can return only one parameter.

SQL ID. SQL authorization ID.

SQLJ. Structured Query Language (SQL) that is embedded in the Java programming language.

SQL path. An ordered list of schema names that are used in the resolution of unqualified references to user-defined functions, distinct types, and stored procedures. In dynamic SQL, the current path is found in the CURRENT PATH special register. In static SQL, it is defined in the PATH bind option.

SQL procedure. A user-written program that can be invoked with the SQL CALL statement. Contrast with *external procedure*.

SQL processing conversation. Any conversation that requires access of DB2 data, either through an application or by dynamic query requests.

SQL Processor Using File Input (SPUFI). A facility of the TSO attachment subcomponent that enables the DB2I user to execute SQL statements without embedding them in an application program.

SQL return code. Either SQLCODE or SQLSTATE.

SQL routine. A user-defined function or stored procedure that is based on code that is written in SQL.

SQL statement coprocessor. An alternative to the DB2 precompiler that lets the user process SQL statements at compile time. The user invokes an SQL statement coprocessor by specifying a compiler option.

SQL string delimiter. A symbol that is used to enclose an SQL string constant. The SQL string delimiter is the apostrophe ('), except in COBOL applications, where the user assigns the symbol, which is either an apostrophe or a double quotation mark (").

SRB. Service request block.

SSI. Subsystem interface (in z/OS).

SSM. Subsystem member (in IMS).

stand-alone. An attribute of a program that means that it is capable of executing separately from DB2, without using DB2 services.

star join. A method of joining a dimension column of a fact table to the key column of the corresponding dimension table. See also *join*, *dimension*, and *star schema*.

star schema. The combination of a fact table (which contains most of the data) and a number of dimension tables. See also *star join*, *dimension*, and *dimension table*.

statement handle. In DB2 ODBC, the data object that contains information about an SQL statement that is managed by DB2 ODBC. This includes information such as dynamic arguments, bindings for dynamic arguments and columns, cursor information, result values, and status information. Each statement handle is associated with the connection handle.

statement string. For a dynamic SQL statement, the character string form of the statement.

statement trigger. A trigger that is defined with the trigger granularity FOR EACH STATEMENT.

static cursor. A named control structure that does not change the size of the result table or the order of its rows after an application opens the cursor. Contrast with *dynamic cursor*.

static SQL. SQL statements, embedded within a program, that are prepared during the program preparation process (before the program is executed). After being prepared, the SQL statement does not change (although values of host variables that are specified by the statement might change).

storage group. A named set of disks on which DB2 data can be stored.

stored procedure. A user-written application program that can be invoked through the use of the SQL CALL statement.

string. See *character string* or *graphic string*.

strong typing. A process that guarantees that only user-defined functions and operations that are defined on a distinct type can be applied to that type. For example, you cannot directly compare two currency types, such as Canadian dollars and U.S. dollars. But you can provide a user-defined function to convert one currency to the other and then do the comparison.

structure. (1) A name that refers collectively to different types of DB2 objects, such as tables, databases, views, indexes, and table spaces. (2) A construct that uses z/OS to map and manage storage on a coupling facility. See also *cache structure*, *list structure*, or *lock structure*.

Structured Query Language (SQL). A standardized language for defining and manipulating data in a relational database.

structure owner. In relation to group buffer pools, the DB2 member that is responsible for the following activities:

- Coordinating rebuild, checkpoint, and damage assessment processing
- Monitoring the group buffer pool threshold and notifying castout owners when the threshold has been reached

subcomponent. A group of closely related DB2 modules that work together to provide a general function.

subject table. The table for which a trigger is created. When the defined triggering event occurs on this table, the trigger is activated.

subpage. The unit into which a physical index page can be divided.

subquery. A SELECT statement within the WHERE or HAVING clause of another SQL statement; a nested SQL statement.

subselect. That form of a query that does not include an ORDER BY clause, an UPDATE clause, or UNION operators.

substitution character. A unique character that is substituted during character conversion for any characters in the source program that do not have a match in the target coding representation.

subsystem. A distinct instance of a relational database management system (RDBMS).

surrogate pair. A coded representation for a single character that consists of a sequence of two 16-bit code units, in which the first value of the pair is a high-surrogate code unit in the range U+D800 through U+DBFF, and the second value is a low-surrogate code unit in the range U+DC00 through U+DFFF. Surrogate pairs provide an extension mechanism for encoding 917 476 characters without requiring the use of 32-bit characters.

SVC dump. A dump that is issued when a z/OS or a DB2 functional recovery routine detects an error.

sync point. See *commit point*.

syncpoint tree. The tree of recovery managers and resource managers that are involved in a logical unit of work, starting with the recovery manager, that make the final commit decision.

synonym. In SQL, an alternative name for a table or view. Synonyms can be used to refer only to objects at the subsystem in which the synonym is defined.

syntactic character set. A set of 81 graphic characters that are registered in the IBM registry as character set 00640. This set was originally recommended to the programming language community to be used for syntactic purposes toward maximizing portability and interchangeability across systems and country boundaries. It is contained in most of the primary registered character sets, with a few exceptions. See also *invariant character set*.

Sysplex. See *Parallel Sysplex*.

Sysplex query parallelism. Parallel execution of a single query that is accomplished by using multiple tasks on more than one DB2 subsystem. See also *query CP parallelism*.

system administrator. The person at a computer installation who designs, controls, and manages the use of the computer system.

system agent. A work request that DB2 creates internally such as prefetch processing, deferred writes, and service tasks.

system conversation. The conversation that two DB2 subsystems must establish to process system messages before any distributed processing can begin.

system diagnostic work area (SDWA). The data that is recorded in a SYS1.LOGREC entry that describes a program or hardware error.

system-directed connection. A connection that a relational DBMS manages by processing SQL statements with three-part names.

System Modification Program/Extended (SMP/E). A z/OS tool for making software changes in programming systems (such as DB2) and for controlling those changes.

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information through and controlling the configuration and operation of networks.

SYS1.DUMPxx data set. A data set that contains a system dump (in z/OS).

SYS1.LOGREC. A service aid that contains important information about program and hardware errors (in z/OS).

T

table. A named data object consisting of a specific number of columns and some number of unordered rows. See also *base table* or *temporary table*.

table-controlled partitioning. A type of partitioning in which partition boundaries for a partitioned table are controlled by values that are defined in the CREATE TABLE statement. Partition limits are saved in the LIMITKEY_INTERNAL column of the SYSIBM.SYSTABLEPART catalog table.

table function. A function that receives a set of arguments and returns a table to the SQL statement that references the function. A table function can be referenced only in the FROM clause of a subselect.

table locator. A mechanism that allows access to trigger transition tables in the FROM clause of SELECT statements, in the subselect of INSERT statements, or from within user-defined functions. A table locator is a fullword integer value that represents a transition table.

table space. A page set that is used to store the records in one or more tables.

table space set. A set of table spaces and partitions that should be recovered together for one of these reasons:

- Each of them contains a table that is a parent or descendent of a table in one of the others.
- The set contains a base table and associated auxiliary tables.

A table space set can contain both types of relationships.

task control block (TCB). A z/OS control block that is used to communicate information about tasks within an address space that are connected to DB2. See also *address space connection*.

TB. Terabyte (1 099 511 627 776 bytes).

TCB. Task control block (in z/OS).

TCP/IP. A network communication protocol that computer systems use to exchange information across telecommunication links.

TCP/IP port. A 2-byte value that identifies an end user or a TCP/IP network application within a TCP/IP host.

template. A DB2 utilities output data set descriptor that is used for dynamic allocation. A template is defined by the TEMPLATE utility control statement.

temporary table. A table that holds temporary data. Temporary tables are useful for holding or sorting intermediate results from queries that contain a large number of rows. The two types of temporary table, which are created by different SQL statements, are the created temporary table and the declared temporary table. Contrast with *result table*. See also *created temporary table* and *declared temporary table*.

Terminal Monitor Program (TMP). A program that provides an interface between terminal users and command processors and has access to many system services (in z/OS).

thread. The DB2 structure that describes an application's connection, traces its progress, processes resource functions, and delimits its accessibility to DB2 resources and services. Most DB2 functions execute under a thread structure. See also *allied thread* and *database access thread*.

threadsafe. A characteristic of code that allows multithreading both by providing private storage areas for each thread, and by properly serializing shared (global) storage areas.

three-part name. The full name of a table, view, or alias. It consists of a location name, authorization ID, and an object name, separated by a period.

time. A three-part value that designates a time of day in hours, minutes, and seconds.

time duration. A decimal integer that represents a number of hours, minutes, and seconds.

timeout. Abnormal termination of either the DB2 subsystem or of an application because of the unavailability of resources. Installation specifications are set to determine both the amount of time DB2 is to wait for IRLM services after starting, and the amount of time IRLM is to wait if a resource that an application requests is unavailable. If either of these time specifications is exceeded, a timeout is declared.

Time-Sharing Option (TSO). An option in MVS that provides interactive time sharing from remote terminals.

timestamp. A seven-part value that consists of a date and time. The timestamp is expressed in years, months, days, hours, minutes, seconds, and microseconds.

TMP. Terminal Monitor Program.

to-do. A state of a unit of recovery that indicates that the unit of recovery's changes to recoverable DB2 resources are indoubt and must either be applied to the disk media or backed out, as determined by the commit coordinator.

trace. A DB2 facility that provides the ability to monitor and collect DB2 monitoring, auditing, performance, accounting, statistics, and serviceability (global) data.

transaction lock. A lock that is used to control concurrent execution of SQL statements.

transaction program name. In SNA LU 6.2 conversations, the name of the program at the remote logical unit that is to be the other half of the conversation.

transient XML data type. A data type for XML values that exists only during query processing.

transition table. A temporary table that contains all the affected rows of the subject table in their state before or after the triggering event occurs. Triggered SQL statements in the trigger definition can reference the table of changed rows in the old state or the new state.

transition variable. A variable that contains a column value of the affected row of the subject table in its state before or after the triggering event occurs. Triggered SQL statements in the trigger definition can reference the set of old values or the set of new values.

tree structure. A data structure that represents entities in nodes, with a most one parent node for each node, and with only one root node.

trigger. A set of SQL statements that are stored in a DB2 database and executed when a certain event occurs in a DB2 table.

trigger activation. The process that occurs when the trigger event that is defined in a trigger definition is executed. Trigger activation consists of the evaluation of the triggered action condition and conditional execution of the triggered SQL statements.

trigger activation time. An indication in the trigger definition of whether the trigger should be activated before or after the triggered event.

trigger body. The set of SQL statements that is executed when a trigger is activated and its triggered action condition evaluates to true. A trigger body is also called *triggered SQL statements*.

trigger cascading. The process that occurs when the triggered action of a trigger causes the activation of another trigger.

triggered action. The SQL logic that is performed when a trigger is activated. The triggered action consists of an optional triggered action condition and a set of triggered SQL statements that are executed only if the condition evaluates to true.

triggered action condition. An optional part of the triggered action. This Boolean condition appears as a WHEN clause and specifies a condition that DB2 evaluates to determine if the triggered SQL statements should be executed.

triggered SQL statements. The set of SQL statements that is executed when a trigger is activated and its triggered action condition evaluates to true. Triggered SQL statements are also called the *trigger body*.

trigger granularity. A characteristic of a trigger, which determines whether the trigger is activated:
- Only once for the triggering SQL statement
- Once for each row that the SQL statement modifies

triggering event. The specified operation in a trigger definition that causes the activation of that trigger. The triggering event is comprised of a triggering operation (INSERT, UPDATE, or DELETE) and a subject table on which the operation is performed.

triggering SQL operation. The SQL operation that causes a trigger to be activated when performed on the subject table.

trigger package. A package that is created when a CREATE TRIGGER statement is executed. The package is executed when the trigger is activated.

TSO. Time-Sharing Option.

TSO attachment facility. A DB2 facility consisting of the DSN command processor and DB2I. Applications that are not written for the CICS or IMS environments can run under the TSO attachment facility.

typed parameter marker. A parameter marker that is specified along with its target data type. It has the general form:

```
CAST(? AS data-type)
```

type 1 indexes. Indexes that were created by a release of DB2 before DB2 Version 4 or that are specified as type 1 indexes in Version 4. Contrast with *type 2 indexes*. As of Version 8, type 1 indexes are no longer supported.

type 2 indexes. Indexes that are created on a release of DB2 after Version 7 or that are specified as type 2 indexes in Version 4 or later.

U

UCS-2. Universal Character Set, coded in 2 octets, which means that characters are represented in 16-bits per character.

UDF. User-defined function.

UDT. User-defined data type. In DB2 UDB for z/OS, the term *distinct type* is used instead of user-defined data type. See *distinct type*.

uncommitted read (UR). The isolation level that allows an application to read uncommitted data.

underlying view. The view on which another view is directly or indirectly defined.

undo. A state of a unit of recovery that indicates that the changes that the unit of recovery made to recoverable DB2 resources must be backed out.

Unicode. A standard that parallels the ISO-10646 standard. Several implementations of the Unicode standard exist, all of which have the ability to represent a large percentage of the characters that are contained in the many scripts that are used throughout the world.

uniform resource locator (URL). A Web address, which offers a way of naming and locating specific items on the Web.

union. An SQL operation that combines the results of two SELECT statements. Unions are often used to merge lists of values that are obtained from several tables.

unique constraint. An SQL rule that no two values in a primary key, or in the key of a unique index, can be the same.

unique index. An index that ensures that no identical key values are stored in a column or a set of columns in a table.

unit of recovery. A recoverable sequence of operations within a single resource manager, such as an instance of DB2. Contrast with *unit of work*.

unit of recovery identifier (URID). The LOGRBA of the first log record for a unit of recovery. The URID also appears in all subsequent log records for that unit of recovery.

unit of work. A recoverable sequence of operations within an application process. At any time, an application process is a single unit of work, but the life of an application process can involve many units of work as a result of commit or rollback operations. In a *multisite update* operation, a single unit of work can include several *units of recovery*. Contrast with *unit of recovery*.

Universal Unique Identifier (UUID). An identifier that is immutable and unique across time and space (in z/OS).

unlock. The act of releasing an object or system resource that was previously locked and returning it to general availability within DB2.

untyped parameter marker. A parameter marker that is specified without its target data type. It has the form of a single question mark (?).

updatability. The ability of a cursor to perform positioned updates and deletes. The updatability of a cursor can be influenced by the SELECT statement and the cursor sensitivity option that is specified on the DECLARE CURSOR statement.

update hole. The location on which a cursor is positioned when a row in a result table is fetched again and the new values no longer satisfy the search condition. DB2 marks a row in the result table as an update hole when an update to the corresponding row in the database causes that row to no longer qualify for the result table.

update trigger. A trigger that is defined with the triggering SQL operation UPDATE.

upstream. The node in the syncpoint tree that is responsible, in addition to other recovery or resource managers, for coordinating the execution of a two-phase commit.

UR. Uncommitted read.

URE. Unit of recovery element.

URID . Unit of recovery identifier.

URL. Uniform resource locator.

user-defined data type (UDT). See *distinct type*.

user-defined function (UDF). A function that is defined to DB2 by using the CREATE FUNCTION statement and that can be referenced thereafter in SQL statements. A user-defined function can be an *external function,* a *sourced function,* or an *SQL function.* Contrast with *built-in function*.

user view. In logical data modeling, a model or representation of critical information that the business requires.

UTF-8. Unicode Transformation Format, 8-bit encoding form, which is designed for ease of use with existing ASCII-based systems. The CCSID value for data in UTF-8 format is 1208. DB2 UDB for z/OS supports UTF-8 in mixed data fields.

UTF-16. Unicode Transformation Format, 16-bit encoding form, which is designed to provide code values for over a million characters and a superset of UCS-2. The CCSID value for data in UTF-16 format is 1200. DB2 UDB for z/OS supports UTF-16 in graphic data fields.

UUID. Universal Unique Identifier.

V

value. The smallest unit of data that is manipulated in SQL.

variable. A data element that specifies a value that can be changed. A COBOL elementary data item is an example of a variable. Contrast with *constant*.

variant function. See *nondeterministic function*.

varying-length string. A character or graphic string whose length varies within set limits. Contrast with *fixed-length string*.

version. A member of a set of similar programs, DBRMs, packages, or LOBs.

A version of a program is the source code that is produced by precompiling the program. The program version is identified by the program name and a timestamp (consistency token).

A version of a DBRM is the DBRM that is produced by precompiling a program. The DBRM version is identified by the same program name and timestamp as a corresponding program version.

A version of a package is the result of binding a DBRM within a particular database system. The package version is identified by the same program name and consistency token as the DBRM.

A version of a LOB is a copy of a LOB value at a point in time. The version number for a LOB is stored in the auxiliary index entry for the LOB.

view. An alternative representation of data from one or more tables. A view can include all or some of the columns that are contained in tables on which it is defined.

view check option. An option that specifies whether every row that is inserted or updated through a view must conform to the definition of that view. A view check option can be specified with the WITH CASCADED CHECK OPTION, WITH CHECK OPTION, or WITH LOCAL CHECK OPTION clauses of the CREATE VIEW statement.

Virtual Storage Access Method (VSAM). An access method for direct or sequential processing of fixed- and varying-length records on disk devices. The records in a VSAM data set or file can be organized in logical sequence by a key field (key sequence), in the physical sequence in which they are written on the data set or file (entry-sequence), or by relative-record number (in z/OS).

Virtual Telecommunications Access Method (VTAM). An IBM licensed program that controls communication and the flow of data in an SNA network (in z/OS).

volatile table. A table for which SQL operations choose index access whenever possible.

VSAM. Virtual Storage Access Method.

VTAM. Virtual Telecommunication Access Method (in z/OS).

W

warm start. The normal DB2 restart process, which involves reading and processing log records so that data that is under the control of DB2 is consistent. Contrast with *cold start*.

WLM application environment. A z/OS Workload Manager attribute that is associated with one or more stored procedures. The WLM application environment determines the address space in which a given DB2 stored procedure runs.

write to operator (WTO). An optional user-coded service that allows a message to be written to the system console operator informing the operator of errors and unusual system conditions that might need to be corrected (in z/OS).

WTO. Write to operator.

WTOR. Write to operator (WTO) with reply.

X

XCF. See *cross-system coupling facility*.

XES. See *cross-system extended services*.

XML. See *Extensible Markup Language*.

XML attribute. A name-value pair within a tagged XML element that modifies certain features of the element.

XML element. A logical structure in an XML document that is delimited by a start and an end tag. Anything between the start tag and the end tag is the content of the element.

XML node. The smallest unit of valid, complete structure in a document. For example, a node can represent an element, an attribute, or a text string.

XML publishing functions. Functions that return XML values from SQL values.

X/Open. An independent, worldwide open systems organization that is supported by most of the world's largest information systems suppliers, user organizations, and software companies. X/Open's goal is to increase the portability of applications by combining existing and emerging standards.

XRF. Extended recovery facility.

Z

z/OS. An operating system for the eServer™ product line that supports 64-bit real and virtual storage.

z/OS Distributed Computing Environment (z/OS DCE). A set of technologies that are provided by the Open Software Foundation to implement distributed computing.

Bibliography

DB2 Universal Database for z/OS Version 8 product information:

- *DB2 Administration Guide*, SC18-7413
- *DB2 Application Programming and SQL Guide*, SC18-7415
- *DB2 Application Programming Guide and Reference for Java*, SC18-7414
- *DB2 Codes*, GC18-9603
- *DB2 Command Reference*, SC18-7416
- *DB2 Common Criteria Guide*, SC18-9672
- *DB2 Data Sharing: Planning and Administration*, SC18-7417
- *DB2 Diagnosis Guide and Reference*, LY37-3201
- *DB2 Diagnostic Quick Reference Card*, LY37-3202
- *DB2 Image, Audio, and Video Extenders Administration and Programming*, SC26-9947 #
- *DB2 Installation Guide*, GC18-7418
- *DB2 Licensed Program Specifications*, GC18-7420
- *DB2 Management Clients Package Program Directory*, GI10-8567
- *DB2 Messages*, GC18-9602
- *DB2 ODBC Guide and Reference*, SC18-7423
- *The Official Introduction to DB2 UDB for z/OS*
- *DB2 Program Directory*, GI10-8566
- *DB2 RACF Access Control Module Guide*, SC18-7433
- *DB2 Reference for Remote DRDA Requesters and Servers*, SC18-7424
- *DB2 Reference Summary*, SX26-3853
- *DB2 Release Planning Guide*, SC18-7425
- *DB2 SQL Reference*, SC18-7426
- *DB2 Text Extender Administration and Programming*, SC26-9948 #
- *DB2 Utility Guide and Reference*, SC18-7427
- *DB2 What's New?*, GC18-7428
- *DB2 XML Extender for z/OS Administration and Programming*, SC18-7431

Books and resources about related products:

APL2®

- *APL2 Programming Guide*, SH21-1072
- *APL2 Programming: Language Reference*, SH21-1061
- *APL2 Programming: Using Structured Query Language (SQL)*, SH21-1057

BookManager® READ/MVS

- *BookManager READ/MVS V1R3: Installation Planning & Customization*, SC38-2035

C language: IBM C/C++ for z/OS

- *z/OS C/C++ Programming Guide*, SC09-4765
- *z/OS C/C++ Run-Time Library Reference*, SA22-7821

Character Data Representation Architecture

- *Character Data Representation Architecture Overview*, GC09-2207
- *Character Data Representation Architecture Reference and Registry*, SC09-2190

CICS Transaction Server for z/OS

The publication order numbers below are for Version 2 Release 2 and Version 2 Release 3 (with the release 2 number listed first).

- *CICS Transaction Server for z/OS Information Center*, SK3T-6903 or SK3T-6957.
- *CICS Transaction Server for z/OS Application Programming Guide*, SC34-5993 or SC34-6231
- *CICS Transaction Server for z/OS Application Programming Reference*, SC34-5994 or SC34-6232
- *CICS Transaction Server for z/OS CICS-RACF Security Guide*, SC34-6011 or SC34-6249
- *CICS Transaction Server for z/OS CICS Supplied Transactions*, SC34-5992 or SC34-6230
- *CICS Transaction Server for z/OS Customization Guide*, SC34-5989 or SC34-6227
- *CICS Transaction Server for z/OS Data Areas*, LY33-6100 or LY33-6103
- *CICS Transaction Server for z/OS DB2 Guide*, SC34-6014 or SC34-6252
- *CICS Transaction Server for z/OS External Interfaces Guide*, SC34-6006 or SC34-6244
- *CICS Transaction Server for z/OS Installation Guide*, GC34-5985 or GC34-6224
- *CICS Transaction Server for z/OS Intercommunication Guide*, SC34-6005 or SC34-6243
- *CICS Transaction Server for z/OS Messages and Codes*, GC34-6003 or GC34-6241
- *CICS Transaction Server for z/OS Operations and Utilities Guide*, SC34-5991 or SC34-6229

© Copyright IBM Corp. 1983, 2008

- *CICS Transaction Server for z/OS Performance Guide*, SC34-6009 or SC34-6247
- *CICS Transaction Server for z/OS Problem Determination Guide*, SC34-6002 or SC34-6239
- *CICS Transaction Server for z/OS Release Guide*, GC34-5983 or GC34-6218
- *CICS Transaction Server for z/OS Resource Definition Guide*, SC34-5990 or SC34-6228
- *CICS Transaction Server for z/OS System Definition Guide*, SC34-5988 or SC34–6226
- *CICS Transaction Server for z/OS System Programming Reference*, SC34-5595 or SC34–6233

CICS Transaction Server for OS/390
- *CICS Transaction Server for OS/390 Application Programming Guide*, SC33-1687
- *CICS Transaction Server for OS/390 DB2 Guide*, SC33-1939
- *CICS Transaction Server for OS/390 External Interfaces Guide*, SC33-1944
- *CICS Transaction Server for OS/390 Resource Definition Guide*, SC33-1684

COBOL:
- *IBM COBOL Language Reference*, SC27-1408
- *Enterprise COBOL for z/OS Programming Guide*, SC27-1412

Database Design
- *DB2 for z/OS and OS/390 Development for Performance Volume I* by Gabrielle Wiorkowski, Gabrielle & Associates, ISBN 0-96684-605-2
- *DB2 for z/OS and OS/390 Development for Performance Volume II* by Gabrielle Wiorkowski, Gabrielle & Associates, ISBN 0-96684-606-0
- *Handbook of Relational Database Design* by C. Fleming and B. Von Halle, Addison Wesley, ISBN 0-20111-434-8

DB2 Administration Tool
- *DB2 Administration Tool for z/OS User's Guide and Reference*, available on the Web at www.ibm.com/software/data/db2imstools/library.html

DB2 Buffer Pool Analyzer for z/OS
- *DB2 Buffer Pool Tool for z/OS User's Guide and Reference,* available on the Web at www.ibm.com/software/data/db2imstools/library.html

DB2 Connect™
- *IBM DB2 Connect Quick Beginnings for DB2 Connect Enterprise Edition*, GC09-4833
- *IBM DB2 Connect Quick Beginnings for DB2 Connect Personal Edition*, GC09-4834
- *IBM DB2 Connect User's Guide*, SC09-4835

DB2 DataPropagator™
- *DB2 Universal Database Replication Guide and Reference*, SC27-1121

DB2 Performance Expert for z/OS, Version 1

The following books are part of the DB2 Performance Expert library. Some of these books include information about the following tools: IBM DB2 Performance Expert for z/OS; IBM DB2 Performance Monitor for z/OS; and DB2 Buffer Pool Analyzer for z/OS.
- *OMEGAMON Buffer Pool Analyzer User's Guide*, SC18-7972
- *OMEGAMON Configuration and Customization*, SC18-7973
- *OMEGAMON Messages*, SC18-7974
- *OMEGAMON Monitoring Performance from ISPF*, SC18-7975
- *OMEGAMON Monitoring Performance from Performance Expert Client*, SC18-7976
- *OMEGAMON Program Directory*, GI10-8549
- *OMEGAMON Report Command Reference*, SC18-7977
- *OMEGAMON Report Reference*, SC18-7978
- *Using IBM Tivoli OMEGAMON XE on z/OS*, SC18-7979

DB2 Query Management Facility (QMF) Version 8.1
- *DB2 Query Management Facility: DB2 QMF High Performance Option User's Guide for TSO/CICS*, SC18-7450
- *DB2 Query Management Facility: DB2 QMF Messages and Codes*, GC18-7447
- *DB2 Query Management Facility: DB2 QMF Reference*, SC18-7446
- *DB2 Query Management Facility: Developing DB2 QMF Applications*, SC18-7651
- *DB2 Query Management Facility: Getting Started with DB2 QMF for Windows and DB2 QMF for WebSphere*, SC18-7449
- *DB2 Query Management Facility: Getting Started with DB2 QMF Query Miner*, GC18-7451
- *DB2 Query Management Facility: Installing and Managing DB2 QMF for TSO/CICS*, GC18-7444
- *DB2 Query Management Facility: Installing and Managing DB2 QMF for Windows and DB2 QMF for WebSphere*, GC18-7448

- *DB2 Query Management Facility: Introducing DB2 QMF*, GC18-7443
- *DB2 Query Management Facility: Using DB2 QMF*, SC18-7445
- *DB2 Query Management Facility: DB2 QMF Visionary Developer's Guide*, SC18-9093
- *DB2 Query Management Facility: DB2 QMF Visionary Getting Started Guide*, GC18-9092

DB2 Redbooks™

For access to all IBM Redbooks about DB2, see the IBM Redbooks Web page at www.ibm.com/redbooks

DB2 Server for VSE & VM
- *DB2 Server for VM: DBS Utility*, SC09-2983

DB2 Universal Database Cross-Platform information
- *IBM DB2 Universal Database SQL Reference for Cross-Platform Development*, available at www.ibm.com/software/data/developer/cpsqlref/

DB2 Universal Database for iSeries

The following books are available at www.ibm.com/iseries/infocenter
- *DB2 Universal Database for iSeries Performance and Query Optimization*
- *DB2 Universal Database for iSeries Database Programming*
- *DB2 Universal Database for iSeries SQL Programming Concepts*
- *DB2 Universal Database for iSeries SQL Programming with Host Languages*
- *DB2 Universal Database for iSeries SQL Reference*
- *DB2 Universal Database for iSeries Distributed Data Management*
- *DB2 Universal Database for iSeries Distributed Database Programming*

DB2 Universal Database for Linux, UNIX, and Windows:
- *DB2 Universal Database Administration Guide: Planning*, SC09-4822
- *DB2 Universal Database Administration Guide: Implementation*, SC09-4820
- *DB2 Universal Database Administration Guide: Performance*, SC09-4821
- *DB2 Universal Database Administrative API Reference*, SC09-4824
- *DB2 Universal Database Application Development Guide: Building and Running Applications*, SC09-4825
- *DB2 Universal Database Call Level Interface Guide and Reference, Volumes 1 and 2*, SC09-4849 and SC09-4850
- *DB2 Universal Database Command Reference*, SC09-4828
- *DB2 Universal Database SQL Reference Volume 1*, SC09-4844
- *DB2 Universal Database SQL Reference Volume 2*, SC09-4845

Device Support Facilities
- *Device Support Facilities User's Guide and Reference*, GC35-0033

DFSMS

These books provide information about a variety of components of DFSMS, including z/OS DFSMS, z/OS DFSMSdfp™, z/OS DFSMSdss, z/OS DFSMShsm, and z/OS DFP.
- *z/OS DFSMS Access Method Services for Catalogs*, SC26-7394
- *z/OS DFSMSdss Storage Administration Guide*, SC35-0423
- *z/OS DFSMSdss Storage Administration Reference*, SC35-0424
- *z/OS DFSMShsm Managing Your Own Data*, SC35-0420
- *z/OS DFSMSdfp: Using DFSMSdfp in the z/OS Environment*, SC26-7473
- *z/OS DFSMSdfp Diagnosis Reference*, GY27-7618
- *z/OS DFSMS: Implementing System-Managed Storage*, SC27-7407
- *z/OS DFSMS: Macro Instructions for Data Sets*, SC26-7408
- *z/OS DFSMS: Managing Catalogs*, SC26-7409
- *z/OS MVS: Program Management User's Guide and Reference*, SA22-7643
- *z/OS MVS Program Management: Advanced Facilities*, SA22-7644
- *z/OS DFSMSdfp Storage Administration Reference*, SC26-7402
- *z/OS DFSMS: Using Data Sets*, SC26-7410
- *DFSMSdfp Advanced Services* , SC26-7400
- *DFSMS/MVS: Utilities*, SC26-7414

DFSORT™
- *DFSORT Application Programming: Guide*, SC33-4035
- *DFSORT Installation and Customization*, SC33-4034

Distributed Relational Database Architecture

- *Open Group Technical Standard*; the Open Group presently makes the following DRDA books available through its Web site at www.opengroup.org
  - *Open Group Technical Standard, DRDA Version 3 Vol. 1: Distributed Relational Database Architecture*
  - *Open Group Technical Standard, DRDA Version 3 Vol. 2: Formatted Data Object Content Architecture*
  - *Open Group Technical Standard, DRDA Version 3 Vol. 3: Distributed Data Management Architecture*

Domain Name System
- *DNS and BIND, Third Edition, Paul Albitz and Cricket Liu, O'Reilly*, ISBN 0-59600-158-4

Education
- Information about IBM educational offerings is available on the Web at http://www.ibm.com/software/sw-training/
- A collection of glossaries of IBM terms is available on the IBM Terminology Web site at www.ibm.com/ibm/terminology/index.html

eServer zSeries®
- *IBM eServer zSeries Processor Resource/System Manager Planning Guide*, SB10-7033

Fortran: VS Fortran
- *VS Fortran Version 2: Language and Library Reference*, SC26-4221
- *VS Fortran Version 2: Programming Guide for CMS and MVS*, SC26-4222

High Level Assembler
- *High Level Assembler for MVS and VM and VSE Language Reference*, SC26-4940
- *High Level Assembler for MVS and VM and VSE Programmer's Guide*, SC26-4941

ICSF
- *z/OS ICSF Overview*, SA22-7519
- *Integrated Cryptographic Service Facility Administrator's Guide*, SA22-7521

IMS Version 8

IMS product information is available on the IMS Library Web page, which you can find at www.ibm.com/ims
- *IMS Administration Guide: System*, SC27-1284
- *IMS Administration Guide: Transaction Manager*, SC27-1285
- *IMS Application Programming: Database Manager*, SC27-1286
- *IMS Application Programming: Design Guide*, SC27-1287
- *IMS Application Programming: Transaction Manager*, SC27-1289
- *IMS Command Reference*, SC27-1291
- *IMS Customization Guide*, SC27-1294
- *IMS Install Volume 1: Installation Verification*, GC27-1297
- *IMS Install Volume 2: System Definition and Tailoring*, GC27-1298
- *IMS Messages and Codes Volumes 1 and 2*, GC27-1301 and GC27-1302
- *IMS Open Transaction Manager Access Guide and Reference*, SC18-7829
- *IMS Utilities Reference: System*, SC27-1309

General information about IMS Batch Terminal Simulator for z/OS is available on the Web at www.ibm.com/software/data/db2imstools/library.html

IMS DataPropagator
- *IMS DataPropagator for z/OS Administrator's Guide for Log*, SC27-1216
- *IMS DataPropagator: An Introduction*, GC27-1211
- *IMS DataPropagator for z/OS Reference*, SC27-1210

ISPF
- *z/OS ISPF Dialog Developer's Guide*, SC23-4821
- *z/OS ISPF Messages and Codes*, SC34-4815
- *z/OS ISPF Planning and Customizing*, GC34-4814
- *z/OS ISPF User's Guide Volumes 1 and 2*, SC34-4822 and SC34-4823

Language Environment
- *Debug Tool User's Guide and Reference*, SC18-7171
- *Debug Tool for z/OS and OS/390 Reference and Messages*, SC18-7172
- *z/OS Language Environment Concepts Guide*, SA22-7567
- *z/OS Language Environment Customization*, SA22-7564
- *z/OS Language Environment Debugging Guide*, GA22-7560
- *z/OS Language Environment Programming Guide*, SA22-7561
- *z/OS Language Environment Programming Reference*, SA22-7562

MQSeries®
- *MQSeries Application Messaging Interface*, SC34-5604

- *MQSeries for OS/390 Concepts and Planning Guide*, GC34-5650
- *MQSeries for OS/390 System Setup Guide*, SC34-5651

National Language Support
- *National Language Design Guide Volume 1*, SE09-8001
- *IBM National Language Support Reference Manual Volume 2*, SE09-8002

NetView®
- *Tivoli NetView for z/OS Installation: Getting Started*, SC31-8872
- *Tivoli NetView for z/OS User's Guide*, GC31-8849

Microsoft ODBC

Information about Microsoft ODBC is available at http://msdn.microsoft.com/library/

Parallel Sysplex Library
- *System/390 9672 Parallel Transaction Server, 9672 Parallel Enterprise Server, 9674 Coupling Facility System Overview For R1/R2/R3 Based Models*, SB10-7033
- *z/OS Parallel Sysplex Application Migration*, SA22-7662
- *z/OS Parallel Sysplex Overview: An Introduction to Data Sharing and Parallelism*, SA22-7661
- *z/OS Parallel Sysplex Test Report*, SA22-7663

The *Parallel Sysplex Configuration Assistant* is available at www.ibm.com/s390/pso/psotool

PL/I: Enterprise PL/I for z/OS
- *IBM Enterprise PL/I for z/OS Language Reference*, SC27-1460
- *IBM Enterprise PL/I for z/OS Programming Guide*, SC27-1457

PL/I: PL/I for MVS & VM
- *PL/I for MVS & VM Programming Guide*, SC26-3113

SMP/E
- *SMP/E for z/OS and OS/390 Reference*, SA22-7772
- *SMP/E for z/OS and OS/390 User's Guide*, SA22-7773

Storage Management
- *z/OS DFSMS: Implementing System-Managed Storage*, SC26-7407
- *MVS/ESA Storage Management Library: Managing Data*, SC26-7397
- *MVS/ESA Storage Management Library: Managing Storage Groups*, SC35-0421
- *MVS Storage Management Library: Storage Management Subsystem Migration Planning Guide*, GC26-7398

System Network Architecture (SNA)
- *SNA Formats*, GA27-3136
- *SNA LU 6.2 Peer Protocols Reference*, SC31-6808
- *SNA Transaction Programmer's Reference Manual for LU Type 6.2*, GC30-3084
- *SNA/Management Services Alert Implementation Guide*, GC31-6809

TCP/IP
- *IBM TCP/IP for MVS: Customization & Administration Guide*, SC31-7134
- *IBM TCP/IP for MVS: Diagnosis Guide*, LY43-0105
- *IBM TCP/IP for MVS: Messages and Codes*, SC31-7132
- *IBM TCP/IP for MVS: Planning and Migration Guide*, SC31-7189

TotalStorage™ Enterprise Storage Server
- *RAMAC Virtual Array: Implementing Peer-to-Peer Remote Copy*, SG24-5680
- *Enterprise Storage Server Introduction and Planning*, GC26-7444
- *IBM RAMAC Virtual Array*, SG24-6424

Unicode
- *z/OS Support for Unicode: Using Conversion Services*, SA22-7649

Information about Unicode, the Unicode consortium, the Unicode standard, and standards conformance requirements is available at www.unicode.org

VTAM
- *Planning for NetView, NCP, and VTAM*, SC31-8063
- *VTAM for MVS/ESA Diagnosis*, LY43-0078
- *VTAM for MVS/ESA Messages and Codes*, GC31-8369
- *VTAM for MVS/ESA Network Implementation Guide*, SC31-8370
- *VTAM for MVS/ESA Operation*, SC31-8372
- *z/OS Communications Server SNA Programming*, SC31-8829
- *z/OS Communicatons Server SNA Programmer's LU 6.2 Reference*, SC31-8810
- *VTAM for MVS/ESA Resource Definition Reference*, SC31-8377

WebSphere® family

- *WebSphere MQ Integrator Broker: Administration Guide*, SC34-6171
- *WebSphere MQ Integrator Broker for z/OS: Customization and Administration Guide*, SC34-6175
- *WebSphere MQ Integrator Broker: Introduction and Planning*, GC34-5599
- *WebSphere MQ Integrator Broker: Using the Control Center*, SC34-6168

z/Architecture™

- *z/Architecture Principles of Operation*, SA22-7832

z/OS

- *z/OS C/C++ Programming Guide*, SC09-4765
- *z/OS C/C++ Run-Time Library Reference*, SA22-7821
- *z/OS C/C++ User's Guide*, SC09-4767
- *z/OS Communications Server: IP Configuration Guide*, SC31-8875
- *z/OS DCE Administration Guide*, SC24-5904
- *z/OS DCE Introduction*, GC24-5911
- *z/OS DCE Messages and Codes*, SC24-5912
- *z/OS Information Roadmap*, SA22-7500
- *z/OS Introduction and Release Guide*, GA22-7502
- *z/OS JES2 Initialization and Tuning Guide*, SA22-7532
- *z/OS JES3 Initialization and Tuning Guide*, SA22-7549
- *z/OS Language Environment Concepts Guide*, SA22-7567
- *z/OS Language Environment Customization*, SA22-7564
- *z/OS Language Environment Debugging Guide*, GA22-7560
- *z/OS Language Environment Programming Guide*, SA22-7561
- *z/OS Language Environment Programming Reference*, SA22-7562
- *z/OS Managed System Infrastructure for Setup User's Guide*, SC33-7985
- *z/OS MVS Diagnosis: Procedures*, GA22-7587
- *z/OS MVS Diagnosis: Reference*, GA22-7588
- *z/OS MVS Diagnosis: Tools and Service Aids*, GA22-7589
- *z/OS MVS Initialization and Tuning Guide*, SA22-7591
- *z/OS MVS Initialization and Tuning Reference*, SA22-7592
- *z/OS MVS Installation Exits*, SA22-7593
- *z/OS MVS JCL Reference*, SA22-7597
- *z/OS MVS JCL User's Guide*, SA22-7598
- *z/OS MVS Planning: Global Resource Serialization*, SA22-7600
- *z/OS MVS Planning: Operations*, SA22-7601
- *z/OS MVS Planning: Workload Management*, SA22-7602
- *z/OS MVS Programming: Assembler Services Guide*, SA22-7605
- *z/OS MVS Programming: Assembler Services Reference, Volumes 1 and 2*, SA22-7606 and SA22-7607
- *z/OS MVS Programming: Authorized Assembler Services Guide*, SA22-7608
- *z/OS MVS Programming: Authorized Assembler Services Reference Volumes 1-4*, SA22-7609, SA22-7610, SA22-7611, and SA22-7612
- *z/OS MVS Programming: Callable Services for High-Level Languages*, SA22-7613
- *z/OS MVS Programming: Extended Addressability Guide*, SA22-7614
- *z/OS MVS Programming: Sysplex Services Guide*, SA22-7617
- *z/OS MVS Programming: Sysplex Services Reference*, SA22-7618
- *z/OS MVS Programming: Workload Management Services*, SA22-7619
- *z/OS MVS Recovery and Reconfiguration Guide*, SA22-7623
- *z/OS MVS Routing and Descriptor Codes*, SA22-7624
- *z/OS MVS Setting Up a Sysplex*, SA22-7625
- *z/OS MVS System Codes* SA22-7626
- *z/OS MVS System Commands*, SA22-7627
- *z/OS MVS System Messages Volumes 1-10*, SA22-7631, SA22-7632, SA22-7633, SA22-7634, SA22-7635, SA22-7636, SA22-7637, SA22-7638, SA22-7639, and SA22-7640
- *z/OS MVS Using the Subsystem Interface*, SA22-7642
- *z/OS Planning for Multilevel Security and the Common Criteria*, SA22-7509
- *z/OS RMF User's Guide*, SC33-7990
- *z/OS Security Server Network Authentication Server Administration*, SC24-5926
- *z/OS Security Server RACF Auditor's Guide*, SA22-7684
- *z/OS Security Server RACF Command Language Reference*, SA22-7687
- *z/OS Security Server RACF Macros and Interfaces*, SA22-7682
- *z/OS Security Server RACF Security Administrator's Guide*, SA22-7683
- *z/OS Security Server RACF System Programmer's Guide*, SA22-7681
- *z/OS Security Server RACROUTE Macro Reference*, SA22-7692
- *z/OS Support for Unicode: Using Conversion Services*, SA22-7649
- *z/OS TSO/E CLISTs*, SA22-7781
- *z/OS TSO/E Command Reference*, SA22-7782

- *z/OS TSO/E Customization*, SA22-7783
- *z/OS TSO/E Messages*, SA22-7786
- *z/OS TSO/E Programming Guide*, SA22-7788
- *z/OS TSO/E Programming Services*, SA22-7789
- *z/OS TSO/E REXX Reference*, SA22-7790
- *z/OS TSO/E User's Guide*, SA22-7794
- *z/OS UNIX System Services Command Reference*, SA22-7802
- *z/OS UNIX System Services Messages and Codes*, SA22-7807
- *z/OS UNIX System Services Planning*, GA22-7800
- *z/OS UNIX System Services Programming: Assembler Callable Services Reference*, SA22-7803
- *z/OS UNIX System Services User's Guide*, SA22-7801

Program Number: 5625-DB2

Printed in USA

GC18-9602-03