

Titan Batch Processing

October 14, 2004



National Institutes of Health
Center for Information Technology
NIH Computer Center
12 South Drive MSC 5607
Bethesda, Maryland 20892-5607

Publication No. CIT172



Contents

1	Introduction.....	1
1.1	Conventions Used in This Manual	2
1.2	Getting Assistance	2
1.3	Ordering Publications	2
2	Defining and Controlling a Batch Job	5
2.1	Job Control Language Overview.....	5
2.1.1	Operating System Overview	5
2.1.2	Structuring Your JCL.....	6
2.2	JOB Statement	6
2.2.1	Restrictions at NIH	7
2.2.2	Name Field.....	7
2.2.3	Operation Field	7
2.2.4	Parameters Field.....	7
	().....	7
	CLASS=a	7
	COND=(code,operator)	8
	GROUP=group	8
	LINES=nnn,action	8
	MSGCLASS=a	8
	MSGLEVEL=(statements,messages)	9
	NOTIFY=userid	9
	PAGES=nnnnnnnn,action.....	9
	PASSWORD=password	9
	programmer-name.....	10
	REGION=region	10
	RESTART=option	10
	TIME=(time).....	10
	TYPRUN=option	12
	USER=userid	13
2.2.5	Examples of JOB Statements.....	13
2.3	JES2 and ThruPut Manager Statements.....	13
2.3.1	JES2 Statements.....	14
	/*AFTER.....	14
	/*BEFORE	15
	/*BOX.....	16
	/*CNTL.....	16
	/*DISCOUNT	17
	/*JCS BATCH	17
	/*JOBPARM.....	18
	/*NETACCT.....	19
	/*OUTPUT.....	20

/*PASS.....	20
/*PRIVATE.....	20
/*ROUTE.....	20
/*WITH.....	21
/*XMIT.....	22
2.3.2 JBS BIND ThruPut Manager Statement.....	22
Example of JBS BIND.....	24
2.3.3 Using UDF To Display Job Status.....	24
2.4 JCLLIB Statement.....	24
2.4.1 Name Field.....	25
2.4.2 Operation Field.....	25
2.4.3 ORDER Parameter.....	25
2.4.4 Example of JCLLIB Statement.....	26
2.5 EXEC Statement.....	26
2.5.1 Restrictions at NIH.....	26
2.5.2 Name Field.....	26
2.5.3 Operation Field.....	27
2.5.4 Parameters Field.....	27
PGM.....	27
PROC and procedure-name.....	28
COND.....	28
PARM.....	30
REGION.....	30
TIME.....	31
2.6 DD Statement.....	32
2.6.1 Restrictions at NIH.....	32
2.6.2 Name Field.....	33
2.6.3 Operation Field.....	34
2.6.4 Parameters Field.....	34
* or DATA.....	34
AMP.....	35
AVGREC.....	35
CHARS.....	35
COPIES.....	35
DATACLAS.....	35
DCB.....	41
DCB Subparameters.....	42
DDNAME.....	46
DEST.....	47
DISP.....	47
DLM.....	47
DSNAME.....	47
DUMMY.....	48
EXPDT.....	48
FCB.....	49

FREE.....	49
HOLD.....	49
LABEL.....	49
MGMTCLAS.....	51
OUTLIM.....	52
OUTPUT.....	52
RETPD.....	52
SPACE.....	52
SYSOUT.....	55
UNIT.....	56
VOLUME.....	57
2.7 OUTPUT JCL Statement.....	57
2.7.1 Restrictions at NIH.....	58
2.7.2 Name Field.....	58
2.7.3 Operation Field.....	59
2.7.4 Parameters Field.....	59
CHARS=chars.....	59
CLASS=class.....	59
COPIES=nnn.....	59
DEFAULT=YES.....	59
DEST=dest.....	59
FCB=fcb.....	59
FORMDEF=fdef.....	59
FORMS=form.....	59
JESDS=ALL.....	59
PAGEDEF=pdef.....	60
ROOM=box.....	60
TITLE=title.....	60
2.7.5 Example of OUTPUT JCL Statement.....	60
2.8 COMMENT Statement.....	60
2.9 Standards for Batch Jobs.....	60
2.9.1 Maximum Limits for Batch Jobs.....	61
2.9.2 Restrictions at NIH.....	62
Box Numbers.....	62
2.9.3 Job Turnaround.....	63
2.10 Creating and Using Data Sets.....	63
2.10.1 Creating Temporary Data Sets on System Scratch Disks.....	63
2.10.2 Creating and Deleting Data Sets.....	64
2.10.3 Creating Permanent or Temporary Data Sets on Tape.....	65
2.10.4 Using Permanent Data Sets.....	66
2.10.5 Using a Passed Data Set.....	67
2.11 Using Cataloged Procedures.....	68
2.11.1 JCLLIB Statement.....	68
2.11.2 Defining Symbolic Parameters in a Cataloged Procedure.....	69

2.11.3	Using Cataloged Procedures	70
2.11.4	Assigning Values to and Nullifying Symbolic Parameters.....	70
2.11.5	Modifying Parameters on the EXEC Statement	70
2.11.6	Overriding or Adding DD Statements	71
2.11.7	Instream Procedures.....	73
2.11.8	INCLUDE Groups	74
2.12	Printing	75
2.12.1	JCL for Printing	75
2.12.2	Output Delivery	75
2.12.3	Forms Available.....	76
2.12.4	Standard Form.....	76
2.12.5	Cut-Sheet Forms	78
2.12.6	Labels.....	80
	Using the WYLABEL procedure.....	81
2.12.7	User-Supplied Forms	81
	Registering and Supplying Forms.....	81
2.12.8	Character Sets	82
	Hexadecimal Tables for Character Sets.....	83
2.12.9	Using Carriage Control.....	83
	Computer Center Forms Control Buffers (FCBs).....	86
	Registering User Forms Control Buffers	87
2.12.10	Table Reference Characters (TRC).....	87
2.12.11	Hardcopy Utility on Titan ISPF	89
2.12.12	Network Printing Services	89
	Macintosh Printing Using PrintShare	89
2.12.13	RJE Workstation Facilities	90
2.12.14	VTAM Printers	90
2.13	Submitting Jobs.....	90
2.13.1	Interactive Job Submission	90
2.13.2	Batch Job Submission from Another Batch Job	90
2.13.3	Batch Job Submission through the Web	90
	Submitting a Single Job Using Web Submit.....	90
	Submitting a Single Job Using TSO Commands.....	91
	Scheduling Recurring Work	91
2.13.4	Batch Job Submission through FTP.....	92
2.13.5	Batch Job Cancellation	92
2.13.6	Job Syntax Checking.....	92
2.13.7	Executing a Job at Another Node	93
2.13.8	Remote Job Submission.....	94
	USER	95
	SIGNON	95
	SIGNOFF.....	95
2.14	Displaying Job Status with IOF.....	95
2.14.1	IOF Job List Menu	96
	Selecting a Job	98

2.14.2	IOF Job Summary Screen	98
	Selecting a Data Set	99
2.14.3	IOF Data Set Browse Screen	99
2.14.4	Saving job output to a data set	99
	Alternative Methods.....	100
2.15	Displaying Job Status with UDF.....	101
2.15.1	Resolving Data Set Contention.....	102
2.16	Displaying and Purging Jobs through the Web	102
2.17	Avoiding Job Turnaround Delays	102
2.18	Understanding Job Output	103
3	Utilities - General Information	107
3.1	IBM Utilities	107
3.2	Restrictions at NIH	108
4	Data Set Utilities.....	111
4.1	The Data Set Wonder Utility	111
4.1.1	EDSUTIL	111
	What it can do	111
	Syntax of Commands.....	111
	EDSUTIL COPY Command.....	112
	EDSUTIL LIST Command.....	114
	EDSUTIL Procedure.....	115
	Examples.....	116
4.2	Backing Up and Recovering Data Sets	116
4.2.1	DSSAVE.....	116
4.2.2	DSGET.....	118
4.2.3	Hints on Backing Up and Recovering Data Sets	120
4.3	Cataloging Data Sets.....	120
4.3.1	VDSUTIL	121
4.3.2	CATDS	123
4.3.3	UNCATDS.....	124
4.4	Comparing Data Sets.....	124
4.4.1	COMPARE	124
4.5	Copying Data Sets.....	125
4.5.1	COPY	125
4.5.2	DSCOPY.....	127
4.5.3	IEBCOPY	129
4.6	Erasing Sensitive Data.....	129
4.6.1	ADSERASE.....	129
4.7	Printing Data Sets	131
4.7.1	DSLIST	131

4.7.2	PRINT	132
4.7.3	WYLABEL	135
4.8	Recalling Data Sets via Batch	136
4.8.1	IEFBR14	136
4.9	Renaming Disk Data Sets	137
4.9.1	DSRENAME	137
4.10	Scratching Disk Data Sets	137
4.10.1	DSSCR	137
4.10.2	DSSCRV	138
4.11	Sorting and Merging Data Sets.....	138
4.11.1	SORT	139
4.11.2	MERGE.....	141
4.12	Direct Access Data Sets	142
4.12.1	DDSSAVE	142
4.12.2	DDSGET.....	144
4.12.3	DDSLIST	146
4.12.4	DDSCOPY	146
4.13	Edit Format Data Sets	148
4.13.1	EDSIN.....	148
4.13.2	EDSOUT.....	150
4.13.3	EDSLIST.....	153
4.13.4	EDSSORT.....	154
4.13.5	EDSSAVE.....	157
4.13.6	EDSGET	158
4.14	Generation Data Sets	161
4.14.1	The Model Data Set Label	162
4.14.2	The Cataloged Index.....	162
4.14.3	Creating a New Generation Data Set.....	163
4.14.4	Deleting a Generation Data Group	163
4.15	Partitioned Data Sets	165
4.15.1	PDSALLOC	165
4.15.2	PDSDEL	166
4.15.3	MEMADD	167
4.15.4	USRPDSL.....	169
4.15.5	PDSPACK.....	170
4.15.6	PDSCOPY.....	171
4.15.7	USRPDSC.....	173
4.15.8	PDSSAVE.....	174
4.15.9	PDSGET	175
4.15.10	PDSLIST.....	177
4.16	VSAM Data Sets.....	181
4.17	Using ISMF.....	181

4.17.1	Accessing ISMF.....	181
4.17.2	Specifying Data Set Names	182
4.17.3	Displaying Data Set Information	182
4.17.4	Determining Total Space	183
4.17.5	Changing Management Class	183
4.17.6	Deleting Specific Backups.....	183
4.17.7	Recovering a Data Set from a Backup.....	184
4.18	Obtaining Data Set Information.....	185
4.18.1	ADSMAP	185
4.19	Obtaining List of Backups for Data Set.....	187
4.19.1	BACKLIST	187
5	Disk Utilities	188
5.1	Guidelines for Maintaining Dedicated Disks.....	188
5.1.1	Technical Coordinator	188
5.1.2	Backup Policy	188
5.1.3	Emergency Procedures.....	188
5.1.4	Compressing a Disk	189
5.1.5	Hints on Maintaining Dedicated Disks	190
5.2	Mapping Disks.....	190
5.2.1	DISKMAP.....	190
5.3	Backing Up and Restoring Disks.....	191
5.3.1	DISKSAVE.....	191
5.3.2	DISKGET	192
5.3.3	ADSRECOV.....	193
6	Tape Utilities.....	197
6.1	Displaying Tape Information.....	198
6.1.1	TAPEINFO	198
6.1.2	VOLLIST	199
6.1.3	TI.....	202
6.1.4	CA-1 Primary Option Menu	202
	Selecting tapes	202
	Modifying the Expiration Date or Scratching a Tape.....	203
6.2	Mapping Tapes.....	203
6.2.1	TAPEMAP	203
7	Transfer Utilities	207
7.1	Transferring Data Between Workstation and Host	207
7.1.1	RMTDSIN.....	207
7.1.2	RMTDSOUT.....	209
7.2	Transferring Data from Host to Host	211
7.2.1	SENDFILE and RCVFILE	211
7.2.2	CONNECT:Direct.....	216

DMBATCH.....	217
8 Job Utilities.....	219
8.1 Checking for JCL Errors.....	219
8.1.1 JCLSCAN.....	219
8.2 Submitting Batch Jobs.....	220
8.2.1 DSSUBMIT Procedure.....	220
8.2.2 DSSUBMIT Program.....	221
8.3 Fetching Jobs.....	221
8.3.1 FETCH.....	221
9 Language Utilities.....	225
9.1 Language Procedures.....	225
9.1.1 FORTRAN.....	225
9.1.2 COBOL.....	225
9.1.3 PL/I.....	226
9.2 Using the Binder.....	226
9.2.1 Estimating Region Size.....	227
9.2.2 Using the Loader.....	228
10 Other Utilities.....	231
10.1 Executing TSO Commands at LOGON.....	231
10.1.1 Entering a Command on the TSO/E LOGON Panel.....	231
10.1.2 Saving Commands in TSLOGON.....	231
10.1.3 Saving Commands in TSLIST.....	231
10.1.4 Moving a TSO Session.....	233
10.1.5 Canceling a TSO Session.....	234
10.2 Executing TSO Commands in Batch.....	234
10.2.1 BATCHTSO.....	234
10.3 Executing WYLBUR Commands.....	235
10.3.1 Scrolling the Screen in WYLBUR.....	235
10.3.2 WYLBUR.....	236
10.3.3 WYLBURL.....	238
10.4 Sending Email from Titan.....	238
10.4.1 SENDMAIL.....	238
10.5 VISION:Builder (MARK IV).....	239
10.6 MAX Utilities.....	240
10.6.1 MAX DATA/UTIL.....	240
10.6.2 MAX/BATCH.....	241
10.6.3 MAX/PDF.....	241
11 Tape Storage.....	243
11.1 Tape Standards.....	243

11.1.1	Recreating Tape Data Sets	243
11.1.2	Bypass Label Processing (BLP).....	244
11.2	CA-1 Tape Management System Features	244
11.3	TMS Error Conditions	246
11.4	Tape Data Set Security	246
11.5	Tape Inventory Management.....	247
11.6	Removal of Tapes.....	247
11.7	Foreign Tape Processing	247
12	Messages and Codes.....	251
12.1	Common ABENDs	251
12.2	ThruPut Manager Messages	255
12.3	TMS ABEND Codes and Messages	256
12.3.1	TMS ABEND Codes for IEFTMS50 Messages	256
12.3.2	TMS ABEND Codes for IEFTMS70 Messages	258
12.3.3	TMS Messages in the JES2 Message Log	259
12.4	ABR Online Messages for Data Set Retrieval	260
INDEX.....		263

1 Introduction

This manual is for users who run batch jobs on Titan at the NIH Computer Center, managed by the Center for Information Technology (CIT). It describes the job control language (JCL) statements required for running batch jobs as well as utility programs available for Titan. Readers of this manual are assumed to have an understanding of elementary JCL.

Most North-style and South-style JCL for submitting batch jobs will work on Titan with minimal changes. For information on changes related to the transition, go to the Titan Transition home page at:

<http://silk.nih.gov/silk/titan>

Additional information on the NIH Computer Center, including accounts, online access, and the services schedule, is available at:

<http://datacenter.cit.nih.gov>

The information in this manual should be used in conjunction with:

- *Titan User's Guide*

as well as other CIT publications and related IBM manuals.

The manual documents a subset of parameters for each JCL statement. For full information about the JCL statements, users should refer to the *IBM OS/390 MVS JCL User's Guide*, GC28-1758 and the *IBM OS/390 MVS JCL Reference*, GC28-1757. The manuals for each system can be ordered from CIT (see below).

Changes that affect job control language, batch processing, and utility programs will be fully tested and announced through:

- *Interface*, the technical newsletter that covers the services and facilities provided by the Center for Information Technology to NIH and other government agencies. The URL for the online version is:

<http://datacenter.cit.nih.gov/interface/>

- CIT-Titan-News, an NIH Listserv list for information about Titan. The URL is:

<http://datacenter.cit.nih.gov/titannews/>

1.1 Conventions Used in This Manual

The following conventions are used:

- Keywords are shown in upper case and must be coded exactly as they appear in the statement syntax.
- Words or letters in lower case represent variables for which specific information must be substituted. The information you substitute must be in upper case with the exception of a programmer name in quotes or a title in quotes.
- If a line of code continues onto the next line, the comma is used as a continuation character and is placed at the end of the line to be continued.
- The variables userid, RACFid, and TSOid are used interchangeably on Titan and refer to your Titan userid. The term userid is used in this manual.

1.2 Getting Assistance

The batch processing facilities and utilities described in this manual receive full support from the Center for Information Technology. Questions should be directed to the NIH Help Desk using one of the following methods:

- Telephone
Call 301--496-4357
- Email
Send email to helpdesk@nih.gov
- Web-based customer support:
<http://support.cit.nih.gov>
- Web-based CIT Service Request:
<http://support.cit.nih.gov/ticket.asp>

1.3 Ordering Publications

The CIT publication ordering services allows users to order hardcopy publications, view publications online, or print publications at the central printers or on their workstation printers. CIT offers many publications and manuals relating to the OS/390 operating system and job control language. The publications service is available on the web at:

<http://publications.cit.nih.gov>

Contact the NIH Help Desk if you require assistance.

Titan users can access IBM BookManager on the OS/390 to view online technical documentation. From the CIT/Titan Primary Option panel, select C for Products, then B for books.

2 Defining and Controlling a Batch Job

This section includes the job control language (JCL) statements that describe the resources needed by a batch job running at this installation. Readers of this section are assumed to have an understanding of elementary JCL. Additional details for any JCL statements may be obtained from the *IBM OS/390 MVS JCL User's Guide*, GC28-1758 and the *IBM OS/390 MVS JCL Reference*, GC28-1757.

Most North-style and South-style JCL for submitting batch jobs will work on Titan with minimal changes. For information on changes related to the transition, go to the Titan Transition home page at:

<http://silk.nih.gov/silk/titan>

2.1 Job Control Language Overview

Job control language (JCL) is the means by which the user tells the Titan operating system and its components which resources are required for a job. Through JCL, the user tells the computer what to do with programs, data sets, and I/O devices.

The major JCL statements of each job—JOB, JES2, Thruput Manager, JCLLIB, EXEC, DD, OUTPUT, and COMMENT—are discussed below.

2.1.1 Operating System Overview

The OS/390 operating system introduces programs to the computing system, initiates their execution, and provides them with all the resources and services necessary for them to do their work. It is made up of a general library of programs that can be tailored to meet many requirements. The installation can select the systems programs that it needs, add its own programs to them, and update existing programs, as needs change.

For illustrative purposes, the programs and routines that compose the operating system are classified as a control program and processing programs. The three main functions of the control program are to:

- Accept and schedule jobs in a continuous flow (job management)
- Supervise on a priority basis each unit of work to be done (task management)
- Simplify retrieval of all data, regardless of the way it is organized and stored (data management)

The processing programs are used to define the work that the computing system is to do and to simplify program preparation. The processing programs consist of:

-
- Language translators (such as the FORTRAN compiler)
 - Service programs (such as the Binder)
 - Problem programs (such as users' programs)

For a description of the operating system currently in use at the NIH Computer Center, see the *Titan User's Guide*.

2.1.2 Structuring Your JCL

Every job must contain a minimum of the following two types of control statements:

- A JOB statement, to mark the beginning of a job and assign a name to the job. The JOB statement is also used to provide certain administrative information, including security, accounting, and identification information. Every job has one and only one JOB statement.
- An EXEC (execute) statement, to mark the beginning of a job step, to assign a name to the step, and to identify the program or procedure to be executed in the step. You can add various parameters to the EXEC statement to customize the way the program executes. Every job has at least one EXEC statement.

In addition to the JOB and EXEC statements, most jobs usually contain:

- One or more DD (data definition) statements, to identify and describe the input and output data to be used in the step. The DD statement may be used to request a previously-created data set, to define a new data set, to define a temporary data set, or to define and specify the characteristics of the output.

For information about using cataloged procedures in a job, see section 2.11.

2.2 JOB Statement

The JOB statement identifies a set of work to the operating system. It contains accounting information as well as parameters for governing the execution of the job.

The JOB statement consists of the characters // in columns 1 and 2 followed by two required fields:

- Name field
- Operation field (the word JOB)

The simplest valid JOB statement on Titan is:

```
//jobname JOB
```

A job that is submitted from a Network Job Entry (NJE) node or Remote Job Entry (RJE) workstation has two additional required fields:

- ,USER=userid
where userid is the Titan userid under which the job will run
- ,PASSWORD=password
where password is the password for the userid

2.2.1 Restrictions at NIH

The following JOB statement parameters are not supported:

```
PERFORM
PRTY
RD
```

2.2.2 Name Field

The job name can be any string of 1 - 8 alphanumeric or national (\$, #, @) characters. The first character must be alphabetic or national (\$, #, @).

Jobs with the same userid and identical names will execute one at a time. To ensure a particular order of execution, see *JES2 Statements* in section 2.3.1.

2.2.3 Operation Field

The operation field contains the keyword JOB and must be preceded and followed by at least one blank.

2.2.4 Parameters Field

You can include the following optional JOB statement parameters:

()

In South-style and North-style job statements, specifies accounting information. Not needed on Titan, but if included, should be null and should follow the keyword JOB.

CLASS=a

Specifies a job execution class. Titan has two job execution classes: X (express) and L (long).

- Class X has a default and maximum CPU time of 100 seconds (1 minute 40 seconds).
- Class L has a default CPU time of 999 seconds (16 minutes 40 seconds) and a maximum CPU time of 999 minutes.

The following rules apply to Titan-style job statements:

-
- The job class can be determined by the CLASS parameter or the TIME parameter.
 - If only CLASS is specified, the job is assigned to the class specified.
 - If only TIME is specified, class is determined by the amount of time specified.
 - If both CLASS and TIME are specified, TIME takes precedence.
 - If neither parameter is specified, the job is assigned to class L.
 - Users who need more than 999 minutes should make arrangements through the NIH Help Desk.

If a job uses a North-style or South-style JOB statement, the class is translated to X or L. See the Titan Transition web pages at <http://silk.nih.gov/silk/titan> for translation information.

COND=(code,operator)

Specifies whether a job will continue to process based on completion codes issued by the job steps. If this parameter is included, it applies to all steps in the job. If the condition specified is met, all subsequent steps are bypassed.

GROUP=group

A RACF group. If a group is not specified, the default group is the group to which your userid belongs.

LINES=nnn,action

Indicates the maximum number of lines, in thousands of lines, to be printed for the job's non-AFP format sysout data sets and the action to be taken by the system if the maximum is exceeded. Output that uses DCF or the AFP drivers is in AFP format and the LINES parameter has no effect. You should use the PAGES parameter instead. Generally, use the LINES parameter unless you are certain you are using DCF or the AFP drivers.

Valid actions are CANCEL, DUMP, and WARNING. If you do not specify an action, the default CANCEL is used.

MSGCLASS=a

Specifies the output class for the JES2 job log. If you use the NIH default for MSGLEVEL, the job log includes the JCL and the system messages.

- A causes the job log to be printed.
- B is reserved for internal use.

-
- H causes the job log to be in HOLD for 7 days to allow it to be viewed and disposed of by the user. If it is still in HOLD after 7 days, it is purged.
 - I causes the job log to be in HOLD for 7 days to allow it to be viewed and disposed of by the user. If it is still in HOLD after 7 days, it is printed.
 - J causes the job log to be immediately discarded by JES2 without processing. (J stands for Junk)
 - M is reserved for SMTP output.
 - T is the same as class H.

If this parameter is omitted, the system defaults to standard MSGCLASS=A. If a message class other than one documented above is specified and the job is run at the NIH, the job log is discarded immediately. If the job is run at another node and output is sent to the NIH, the job log is discarded within an hour or two.

See also SYSOUT classes in the parameters on the DD statement (section 2.6.4).

MSGLEVEL=(statements,messages)

Indicates which statements and messages are to be written in the output listing.

MSGLEVEL=(1,1) is the default and should not be overridden. It tells the system to print all control statements and all allocation and termination messages. If you change the default and the job fails, CIT cannot help you analyze the problem.

NOTIFY=userid

Specifies that a job completion message be transmitted to a TSO user.

PAGES=nnnnnnn,action

Indicates the maximum number of pages to be printed for the job's AFP format sysout data sets and the action to be taken by the system if the maximum is exceeded. The value for nnnnnnn is 0 through 99999999. Output that uses DCF or the AFP drivers is in AFP format. For formats that are not AFP, the PAGES parameter has no effect and you should use the LINES parameter instead.

Valid actions are CANCEL, DUMP, and WARNING. If you do not specify an action, the default CANCEL is used.

PASSWORD=password

Identifies the current RACF password. See the *Titan User's Guide* for the rules regarding passwords.

A RACF password is required if the USER= parameter is used and the specified userid is not that of the user submitting the job.

programmer-name

Follows JOB or () and must be in quotes if it contains spaces or special characters.

REGION=region

Specifies the amount of storage to be allocated to a job and should be used whenever the storage required by a job exceeds or is significantly less than the default. If this parameter is omitted, the default region of 32M is used above the line and 4M below the line.

If the JOB statement specifies a REGION parameter, REGION parameters on EXEC statements will be ignored. It is better practice to specify the REGION parameter on the EXEC statements than on the JOB statement.

Replace *region* with one of the following:

nnnnK where *nnnn* represents multiples of K (K=1024 bytes)

nnnnM where *nnnn* represents multiples of M (M=1024K)

If your job fails because of region, increase the region and rerun the job.

RESTART=option

Allows restarting at the beginning of a step. Replace *option* with one of the following:

* indicates that execution is to be restarted at the first job step.

stepname specifies that execution is to be restarted at the named job step.

stepname.procstepname specifies that execution is to be restarted at a cataloged procedure step. *Stepname* is the name of the job step that calls the cataloged procedure, and *procstepname* is the name of the procedure step.

The CHECKID subparameter is not supported.

TIME=(time)

Specifies the maximum amount of execution CPU time required by the job.

Replace *time* with one of the following:

mmm,ss

mmm
,ss

where

mmm represents minutes

ss represents seconds

The maximum number of minutes is 999 if no seconds are specified, 998 otherwise; the maximum number of seconds is 59.

999 minutes is equivalent to 59,940 seconds, or 16 hours and 39 minutes. If more time is needed for a job, contact the NIH Help Desk to make special arrangements.

The **job class** can be determined by the CLASS parameter or the TIME parameter.

The following rules apply:

- **Class X (express)** has a default and maximum CPU time of 100 seconds (1 minute 40 seconds).
- **Class L (long)** has a default CPU time of 999 seconds (16 minutes 39 seconds) and a maximum CPU time of 999 minutes.
- If only CLASS is specified, the job is assigned to the class specified and the default time for the class is used.
- If only TIME is specified, class is determined by the amount of time specified. Jobs that specify no more than 1 minute 40 seconds (TIME=1,40) are assigned to **class X**. All others are assigned to **class L**.
- If both CLASS and TIME are specified, TIME takes precedence.
- If neither parameter is specified, the job is assigned to **class L**.
- Users who need more than 999 minutes should make arrangements through the NIH Help Desk.

If your job specifies TIME and requires less than the time specified, you are charged only for the time used.

If your job requires more than the time specified, it will abend.

In deciding how much time to specify, consider these points:

- Titan reports the actual time used, so after running a job once, you know how long it should take.

- On the first execution, try to estimate an amount of time that is generous enough to allow the job to complete but short enough to stop looping.
- The amount of time specified does not affect scheduling of a particular job within its class. However, accurate time specification for each job enables Titan to schedule its resources more efficiently for all jobs.

TYPRUN=option

Specifies special JES2 processing. Replace *option* with one of the following:

COPY Requests that JES2 copy the input job stream, as submitted, directly to a sysout data set and schedule the sysout data set for output processing. The system does not schedule the job for execution. The class of this sysout data set is the same as the message class of the job and is controlled by the JOB MSGCLASS parameter.

SCAN Requests that the system scan this job's JCL for syntax errors without executing the job or allocating devices. This parameter asks the system to check for:

- Invalid spelling of parameter keywords and some subparameter keywords
- Invalid characters
- Unbalanced parentheses
- Misplaced positional parameters on some statements
- Invalid syntax on JCL statements in cataloged procedures invoked by any scanned EXEC statements

The system does not check for misplaced statements, for invalid syntax in JCL subparameters, or for parameters or subparameters that are inappropriate together.

TYPRUN=SCAN checks the JCL only through the converter, not the interpreter. The difference is that the converter basically checks all expressions to the LEFT of an equal sign plus SOME expressions to the right of an equal sign (and issues messages that start with IEF), while the interpreter checks all expressions to the RIGHT of an equal sign (and issues messages that start with IEF). For example, a data set name containing a qualifier that exceeds eight characters, such as

```
DSNAME=L9755TB.JCL.TEST19970103
```

would NOT be flagged by TYPRUN=SCAN but would be caught by the interpreter.

The HOLD and JCLHOLD options are not supported.

USER=userid

Specifies the userid under which the job is run. If this parameter is omitted, the job inherits the userid of the user or batch job that submitted the job. If the specified userid is not that of the user submitting the job, the PASSWORD parameter must be used.

The userid and password are required for all jobs submitted from an NJE node or RJE node.

2.2.5 Examples of JOB Statements

The following examples illustrate the use of the JOB statement, and the use of various parameters to limit the job's use of resources. This should be done whenever possible, particularly when testing new programs.

This following example is a basic JOB statement:

```
//MYJOB JOB ,SMITH
```

Another example of a Titan JOB statement is:

```
//MYJOB JOB ( ),DASHBROOK,NOTIFY=$JDA,LINES=123, . . . .
```

The following example of a JOB statement can use up to 15 seconds of CPU time and will be placed into the OUTPUT HOLD queue when the job completes:

```
//MYJOB JOB ( ), 'J SMITH PROGRAMMER' ,  
// TIME=( ,15 ),MSGCLASS=H,USER=userid,PASSWORD=password
```

2.3 JES2 and ThruPut Manager Statements

The **Job Entry Subsystem (JES2)** is an extension of the operating system that controls the processing of all jobs. Titan uses the JES2 Multi-Access Spool facility, which allows jobs to execute and print on any Titan CPU. Jobs can be submitted to JES2 from ISPF, from the Web (<http://titan.nih.gov/tsocmd>), from RJE workstations, from databases, or from teleprocessing application systems.

When jobs are submitted, JES2 stores them on a direct access device called the JES2 Spool Disk. Jobs from the spool disk are selected for execution as the required resources become available. All print output is intercepted by JES2 and stored on output spool disks. After job completion, JES2 processes the output when the appropriate output device is available.

ThruPut Manager is an MVS JES2 automation tool, designed to improve the management of batch processing. ThruPut Manager analyzes the batch job stream (or any selected portion) to determine which rules apply.

You can use JES2 and ThruPut Manager statements in your JCL. The statements should be placed after the JOB statement and before the first EXEC statement. If you use statements that give conflicting instructions, the last instruction is the one that will be followed.

2.3.1 JES2 Statements

JES2 control statements contain the characters `/*` in columns 1 and 2 and the JES2 command beginning in column 3.

/*AFTER

The AFTER statement allows you to indicate that a job (Job A) must run after a particular job (Job B). Job B must be in the system at the time the determination is made. If Job B is not in the system, it is assumed that it has already run. Jobs A and B must either run under the same userid or must each use the JCS BATCH statement to assign the same batch name to the jobs.

The format is:

```
/*AFTER jobname [ ,JOB=jobnumber ]
```

where:

jobname Specifies the name of the job that must run before this job can execute. If a job with that name is not found, it is assumed that it has run.

The scope of AFTER applies only to jobs with the same batch name. See the JCS BATCH statement below.

jobnumber Allows you to be specific about a JES2 job number when there are multiple jobs with the same job name. This is only applicable in situations where the named job has already been submitted and you know its JES2 job number.

The ",JOB=jobnumber" portion of the statement is optional.

A job may have multiple AFTER statements. They must all be satisfied before the job will be selected for execution. Because this control statement prevents the system from selecting jobs in their natural sequence, job turnaround may take longer.

Example of AFTER Statement

The following example shows how to prevent job JOB2 from executing until job JOB1 has ended:

```
//JOB1 JOB ...  
// EXEC ...  
  
//JOB2 JOB ...  
/*AFTER JOB1  
// EXEC ...
```

/*BEFORE

The BEFORE statement allows you to indicate that a job (Job A) must run before a particular job (Job B). If Job B is in the system, it is held until Job A (which has the BEFORE statement) completes execution. If Job B is not found, no action is taken. Jobs A and B must either run under the same userid or must each use the JCS BATCH statement to assign the same batch name to the jobs.

The format is:

```
/*BEFORE jobname[ ,JOB=jobnumber ]
```

where:

jobname Specifies the name of the job that is to be held until this job runs. If a job with that name is not found, no action is taken.

The scope of BEFORE applies only to jobs with the same batch name. See the JCS BATCH statement below.

jobnumber Allows you to be specific about a JES2 job number when there are multiple jobs with the same job name. This is only applicable in situations where the named job has already been submitted and you know its JES2 job number.

The ",JOB=jobnumber" portion of the statement is optional.

A job may have multiple BEFORE statements. All of the named jobs that are found in the system are held until this job completes execution. Because this control statement prevents the system from selecting jobs in their natural sequence, job turnaround may take longer.

Example of BEFORE Statement

The following example shows how to prevent job JOB1 from executing until job JOB2 has ended:

```
//JOB1 JOB ...  
/*+JBS BIND HOLD6PM
```

```
// EXEC ...  
  
//JOB2 JOB ...  
/*BEFORE JOB1  
// EXEC ...
```

Note that, in order for this to work, JOB1 must have something to prevent it from beginning execution as soon as it is submitted. In this example, JOB1 will be held to execute after 6:00 pm.

/*BOX

The BOX statement is used to direct local job output to an alternate box. If this statement is omitted, the job is placed in your default output box. The format is:

```
/*BOX nnn
```

or

```
/*BOX Pnnn
```

where:

nnn is the desired box number at the NIH main campus

Pnnn is the desired box number at the Parklawn site

If you specify both BOX and JOBPARM ROOM, the last one specified takes precedence. For more information about box numbers, see section 2.12.2.

/*CNTL

The CNTL statement permits users to designate groups of jobs to be prevented from running when the current job runs.

There are three CNTL statements which may be included in a job:

```
/*CNTL controlname,EXC  
/*CNTL controlname,SHR  
/*CNTL controlname,EXC,DRAIN
```

where

controlname is a control name that identifies a resource. It can be up to 8 characters with no blanks or commas.

EXC grants exclusive use of the resource and ensures that no other job with the same control name will be executed simultaneously.

SHR grants shared use of the resource and enables jobs with the same control name using the SHR parameter to execute but not jobs with the same control name using the EXC parameter. SHR is the default.

DRAIN after all jobs using the resource in shared mode and submitted earlier than the EXC job finish, initiates the job.

DRAIN is useful for jobs that cannot get the resource because other jobs keep entering the job queue in shared mode. It prevents jobs that enter the job queue after the selected EXC job from displacing the EXC job.

The CNTL statements are typically used to regulate the execution of jobs which access a private resource shared by many users. The resource may be a tape containing one or more data sets or a disk data set. A CNTL statement using *controlname* to identify the resource is included in each job that accesses the resource. Jobs that can share the resource (e.g., reading a program library) contain the SHR parameter. Jobs that must have exclusive use of the resource (e.g., updating a program library or using a tape) contain the EXC parameter.

The coordinator of the resource should contact (e.g., via a mailing list) the individuals who access the data to coordinate updates and other accesses requiring exclusive control and to announce the controlname. The coordinator can determine who uses the controlname. If a resource coordinator reports to the NIH Help Desk that jobs using the controlname without permission are causing delays, the job owners will be contacted and their jobs cancelled.

AFTER and BEFORE should not be used in combination with CNTL.

/*DISCOUNT

The DISCOUNT statement will be converted to the ThruPut Manager OVERNITE statement. See OVERNITE in section 2.3.2.

/*JCS BATCH

The JCS BATCH ThruPut Manager control statement is used to assign the same batch name to a group of jobs. It permits jobs run under different userids to be associated so that they can use the BEFORE and AFTER control statements (see these statements above).

The format is:

```
/*JCS BATCH name[ ,SEQUENCE ]
```

where:

name	Specifies the batch name. For naming conventions, see below.
SEQUENCE	When coded, this optional subparameter ensures that, within the named batch, this job will be executed in the sequence in which it was read in. SEQUENCE may be abbreviated to SEQ. See "Forcing First-in, First-run" below.

Batch Name Conventions

The batch name may consist of one or two parts, each 8 characters long. The first character of each part must be alphabetic or national (@, #, or \$); the remaining characters can be alphabetic, numeric, national or hyphen. If a two part name is specified, the parts must be separated by a period.

- If the job does not contain a /*JCS BATCH statement, ThruPut Manager assigns the userid the job runs under as the batch name.
- For two-part names, please make the first part a userid belonging to one of the users to increase the chance that the name will be unique.

Forcing First-in, First-run

Coding the ",SEQUENCE" subparameter on the /*JCS BATCH statement will force all jobs with the same batch name to be executed in the sequence in which they were submitted. For example, including the following statement will insure that all jobs including it will run in the order they are submitted regardless what userid they run under:

```
/*JCS BATCH DOUG.UPDATE,SEQUENCE
```

Only one /*JCS BATCH statement is allowed per job.

/*JOBPARM

The JOBPARM statement permits the user to specify job parameters to JES2 via one or more keywords and values.

JOBPARM statement parameter values will supersede the corresponding specific or default value on the JOB statement or on any previous JOBPARM statement. Any number of parameters may be specified on a single JOBPARM statement and any number of JOBPARM statements may be used within a given job.

The format is:

```
/*JOBPARM keyword,keyword,...
```

The following table shows the most common keywords and values.

JOBPARM Keywords and Values

Long Form	Short Form	Purpose and Permissible Values
COPIES=nnn	N=nnn	Specifies the number of copies to be printed. The numeric values range from 1 to 255. The COPIES parameter is ignored and only one copy is produced if any of the following is true: <ul style="list-style-type: none"> • FREE=CLOSE is coded on the DD statement for the output data set. • HOLD=YES is coded on any SYSOUT DD statement in the job. • The output class of the SYSOUT data set is a held class, and the message class is also a held class. The message class is specified in the JOB statement MSGCLASS parameter.
FORMS=ffff	F=ffff	Specifies form type to be used for printed output. A one-to-four character alphanumeric value.
LINECT=kkk	K=kkk	Maximum number of lines to be printed on each page. The numeric values range from 0 to 255.
LINES=mm	L=mm	Estimated number of output lines to be printed, in thousands. The numeric values range from 0 to 999.
RESTART=y	E=y	Specifies that the job can be restarted from the beginning following a system crash.
ROOM=box ROOM=Pbox	R=box R=Pbox	Specifies an output box. <i>Box</i> specifies a box on central campus and <i>Pbox</i> specifies an output box at Parklawn.

If you specify both BOX and JOBPARM ROOM, the last one specified takes precedence. For more information about box numbers, see section 2.12.2.

See Printing (section 2.12) for information about forms and printed output.

/*NETACCT

The NETACCT statement is used in jobs that execute on a non-NIH node and then transmit SYSOUT to Titan for printing. This statement is required so that the SYSOUT may be associated with its owner at CIT and distributed correctly. The format is:

```
/*NETACCT userid
```

where *userid* is the Titan userid associated with a user. Before the job is accepted for printing on Titan, you must call the NIH Help Desk to establish a facility class with the following information:

- The node name of the originating node
- The userid at that originating node

- One or more associated valid Titan userids

If a job does not contain a NETACCT statement or the facility class is not established, the job will be rejected and returned to the sending node. See section 2.13.7 for examples of jobs using the NETACCT statement.

/*OUTPUT

IBM recommends that you do not use the JES2 OUTPUT statement. Use the OUTPUT JCL statement (section 2.7) instead because the OUTPUT JCL statement has enhanced output processing capabilities.

/*PASS

CIT recommends that you do not use the JES2 PASS statement. Use the PASSWORD parameter on the JOB statement (section 2.2.4) instead.

All Titan users are required to provide a RACF password when submitting a batch job to prevent unauthorized use of account registration codes.

/*PRIVATE

The PRIVATE statement causes the word *PRIVATE* to be printed on the header and trailer pages of the job's printout. The format is:

```
/*PRIVATE
```

This statement should be used for all printed reports that contain personal, proprietary, or other sensitive data. Distribution of all locally produced reports produced with this designation is controlled by Titan. Contact the NIH Help Desk for more information.

/*ROUTE

The ROUTE statement is used to direct print output to a location different from the submitting location. An older form of the statement is also used to specify other resources. See details below.

The format for printing is:

```
/*ROUTE PRINT destination
```

where

destination	specifies the default destination of all print output data sets produced by the job. This specification can be overridden for any particular output data set by specifying a destination on the output DD statement defining it. See the DEST parameter of the DD statement in section 2.6.4.
-------------	---

RJE workstations do not need a ROUTE statement. All output is automatically routed back to the site that submitted the job unless otherwise specified.

Example:

```
/*ROUTE PRINT RMT23
```

This statement will direct the print output to remote job entry workstation number 23.

Using /*ROUTE XEQ

If you are using a data set from the South system that contains the statement:

```
/*ROUTE XEQ resource
```

the statement is translated to the ThruPut Manager statement:

```
//*+JBS BIND agent
```

where the agent specifies a resource.

If your job has the following statement:

```
/*ROUTE XEQ VTAM1
```

you **must change** the statement to:

```
/*ROUTE XEQ CONNECT
```

The resource VTAM1 is not supported on Titan. The CONNECT agent will route the job to the system running CONNECT:Direct.

/*WITH

The WITH statement indicates that a job must run with another concurrently running job. The format is:

```
/*WITH jobname[ ,NOSYSAFF]
```

where:

jobname Specifies the name of the job that must be running before this job can execute. Once the required job begins execution, this job is selected for execution as soon as an initiator with the appropriate class becomes available.

NOSYSAFF Specifies that the jobs do not have to run on the same system

within the JES2 complex.

Examples:

```
/*WITH PAYROLL
```

The job can run only if a job named PAYROLL is running on the same JES2 complex member.

```
/*WITH IMSMAINT ,NOSYSAFF
```

The job can run if a job named IMSMAINT is running anywhere in the JES2 complex.

The WITH statement is subject to the following restrictions:

- If more than one WITH statement is found in the JCL, the last one encountered takes effect.
- If a job references itself in the WITH statement, it is an error and the job fails.
- When two jobs reference each other, neither job will be selected for execution until one of the jobs is released manually.

/*XMIT

The XMIT statement identifies a job that will be executed at a node other than the originating node. A job card image for the originating site must precede the XMIT statement, and a job card image for the executing site must follow the XMIT statement. The format is:

```
/*XMIT nodename DLM=xx
```

where

nodename is the name of the node to which the job is to be transmitted.
For a list of valid node names, see section 7.2.1

xx is a delimiter such as \$\$ which is coded in columns 1-2 of the card image that terminates the statements to be transmitted.

If the DLM=xx parameter is omitted from the XMIT statement, the delimiter /* is used.

2.3.2 JBS BIND ThruPut Manager Statement

The JBS BIND ThruPut Manager control statement is used to specify a resource.

The statement contains the characters // *+ in columns 1 through 4 and the JBS BIND command in columns 5 through 12. The command is followed by an agent that specifies the resource.

The format is:

```
// *+JBS BIND agent
```

You can include up to 4 agents in one JBS BIND statement. Separate the agents with a space. The job will run if any one agent is turned on.

If you want your job to run only if all named agents are turned on, place each agent in a separate JBS BIND statement.

Agents are turned on at a specified time or event and off at a specified time or event.

The following agents are associated with the time of day. Jobs naming an agent run only when the agent is on.

Agent	When Turned On	When Turned Off
HOLD6PM	6:00 pm daily	7:00 am daily
HOLD7PM	7:00 pm daily	7:00 am daily
HOLD8PM	8:00 pm daily	7:00 am daily
HOLD9PM	9:00 pm daily	7:00 am daily
HOLD10PM	10:00 pm daily	7:00 am daily
HOLD11PM	11:00 pm daily	7:00 am daily
HOLD12AM	12:00 am (midnight) daily	7:00 am daily
HOLD2AM	2:00 am daily	7:00 am daily
HOLD4AM	4:00 am daily	7:00 am daily
HOLD5AM	5:00 am daily	7:00 am daily
HOLD6AM	6:00 am daily	7:00 am daily
OVERNITE	5:00 pm daily	7:00 am daily
LATE	10:00 pm daily	7:00 am daily
WEEKEND	5:00 pm Friday	7:00 am Monday

The following agents are associated with specific applications. Jobs naming an agent run only when the agent is on.:

Agent	When Turned On	When Turned Off
ADAOFM	When ADA533 is up	When ADA533 is down
DB2	When DB2PROD is up	When DB2PROD is down
DB2DEV	When DB2DEV is up	When DB2DEV is down
DB2PROD	When DB2PROD is up	When DB2PROD is down
DB2DSNX	When DB2DSNX is up	When DB2DSNX is down
DELAY204	6:30 pm daily	8:30 pm daily
DFM	8:00 pm daily	2:00 am daily

DWNIH	When DWNIH is up	When DWNIH is down
HOLD533	After the ADA533 backup	6:00 am daily
HOLDCICS	When CICS is down.	When CICS is up.
HOLD204	After MODEL204 backups.	6:00 am daily
IMS	When IMS is up.	When IMS is down.
IMSBATCH	When IMS is down.	When IMS is up.
IMSTEST	When IMSTEST is up.	When IMSTEST is down.
M204	When MODEL 204 is up.	When MODEL 204 is down.

CONNECT	routes a job to the system running CONNECT:Direct
ORACLE	routes a job to the system running Oracle

Notes:

Jobs run running between 5 PM and 7AM or on weekends are charged at discount rates. For more information on discount pricing, see the *Titan User's Guide*.

Federal holidays are treated the same as weekdays.

Example of JBS BIND

```
// *+JBS BIND OVERNITE
```

```
// *+JBS BIND WEEKEND
```

2.3.3 Using UDF To Display Job Status

The User Display Facility (UDF) allows users to display information about jobs processed by ThruPut Manager, including whether a job is delayed because it is waiting for a resource to become available. For information about UDF, see section 2.14.4.

2.4 JCLLIB Statement

The JCLLIB statement specifies procedure libraries to be searched before the standard system libraries are searched. The statement is optional.

The JCLLIB statement consists of the characters // in columns 1 and 2 followed by four fields:

- Name
- Operation (the word JCLLIB)
- Keyword parameter (the word ORDER)
- Comments

The format is:

```
//[name] JCLLIB ORDER=(libraries) comments
```

The JCLLIB statement must be placed after the JOB statement and before the first EXEC statement.

2.4.1 Name Field

The name is 1 to 8 alphanumeric or national (\$, #, @) characters and the first character must be alphabetic or national.

The name is optional.

2.4.2 Operation Field

The operation field consists of the characters JCLLIB and must be preceded and followed by at least one blank. It can begin in any column.

2.4.3 ORDER Parameter

The JCLLIB statement contains one keyword parameter:

```
ORDER=(userid.library1,userid.library2,...)
```

where:

library	Specifies the name of a library to be searched. The maximum number of libraries that may be specified is 15. You can specify private libraries, system procedure libraries, and installation-defined procedure libraries. The system searches the libraries in the order in which you specify them, before it searches any unspecified default system procedure libraries.
---------	--

Do not specify a library that is a temporary data set (&&dsname), partitioned data set if a member name is included, or relative generation number for a generation data group (GDG).

Titan does not permit users to store their procedures in any of the system procedure libraries. Users may establish a private procedure library for their procedures. The library should be maintained as a partitioned data set (PDS) and all rules for creating and maintaining a PDS apply. The following rules also apply:

- The library must reside on a dedicated or public disk volume.
- The logical record length must be 80 and record format must be F or FB.
- The library must be cataloged.

2.4.4 Example of JCLLIB Statement

The following JCLLIB statement identifies a cataloged user procedure library:

```
//... JOB ...  
// JCLLIB ORDER=(userid.dsname)  
//STEP1 EXEC ...
```

2.5 EXEC Statement

The EXEC statement identifies the program or procedure to be executed. This statement must be the first JCL statement of each step in a job.

The EXEC statement consists of the characters // in columns 1 and 2 followed by four fields:

- Name
- Operation (the word EXEC)
- Parameters
- Comments

The format is:

```
//stepname EXEC parameters comments
```

2.5.1 Restrictions at NIH

The following EXEC statement parameters are not supported:

```
ACCT  
DPRTY  
PERFORM  
RD
```

2.5.2 Name Field

The step name is optional. If supplied, it must begin in column 3 and consist of 1 to 8 alphanumeric or national (\$, #, @) characters. The first character must be alphabetic or national.

It must be specified if:

- Backwards references are made to the step.

- The step is to be referenced by the RESTART parameter on the JOB statement.
- The step is part of a multi-step procedure and must be specifically referenced. A specific reference to the stepname may be required to add or override DD statements to the step, or to add or override parameters on the EXEC statement within the procedure.

If the step name is omitted, one or more spaces must separate the // and the keyword EXEC.

2.5.3 Operation Field

The operation field contains the keyword EXEC and must be preceded and followed by at least one blank.

2.5.4 Parameters Field

At least one parameter, either PGM, PROC, or a procedure name, must be specified. All other parameters are optional, may appear in any order, and are separated by commas. The most common parameters are:

PGM

Names the program to be executed.

PGM=name

Replace *name* with one of the following:

- program-name
- *.stepname.ddname
- *.stepname.procstepname.ddname

where:

program-name	1 - 8 alphanumeric or \$, #, @ characters identifying the PDS member containing program
stepname	step that contains the DD statement. If the DD statement is in a procedure, use <i>procnamestep</i> to identify the step.
ddname	name of the DD statement being referenced
procstepname	step in named procedure

The program must be in load module form in a partitioned data set.

Examples

```
//LOAD EXEC PGM=HEWL
//SYSLMOD DD DSNAME=&&GOSET(MAIN),UNIT=SYSDA,
// DISP=(,PASS,DELETE),
// SPACE=(CYL,(2,1,1),RLSE)
//GO EXEC PGM=* .LOAD.SYSLMOD,REGION=4M
```

PROC and procedure-name

Names the cataloged or instream procedure to be invoked and executed.

PROC=procedure-name

The keyword PROC may be omitted; any name that appears in this position not preceded by the keyword PGM is assumed to be a procedure name.

The procedure must contain EXEC statements for the program to be executed. If an EXEC statement specifies a multi-step procedure, the keyword parameters should generally be qualified to specify the steps to which they apply.

Examples

```
//COST EXEC PRINT
//STEP#2 EXEC FORTCLG, PARM.LKED=XREF, REGION.GO=4M
```

COND

Tests return codes or ABENDS of previous steps to determine if current step is to be executed. You can also test using the IF/THEN/ELSE/ENDIF constructs.

COND=(code,operator)
COND=abend-test

where:

code must be a number from 0 through 4095. This number is compared with the return codes issued in all previous steps.

operator describes the type of comparison to be made. The operator may be any one of the following:

GT	greater than
GE	greater than or equal to
EQ	equal to
LT	less than

LE less than or equal to
NE not equal to

The operator is used to compare the code to the return code from each previous step. If the comparison is TRUE, then the current step (with the COND parameter) is bypassed.

abend-test may be either EVEN or ONLY.

EVEN specifies that the job step is to be executed even if a previous job step has ABENDED.

ONLY specifies that the job step is to be executed only if a previous job step has ABENDED.

Specific return code values (and their meanings) depend upon the program being executed. The compilers and utility programs provided by the NIH Computer Center generally adhere to the following convention:

0	Execution was successful.
4	A warning or informational message was generated. Although the message should be examined, processing was generally successful.
8	An error was encountered.
12	An error was encountered.
16	An error was encountered.

- Up to 8 different return code tests may be made for each job step. If any of the tests are satisfied, the step is bypassed.
- If only one return code is tested, the outer parentheses are not needed.
- If the COND parameter is coded on both the JOB statement and on one or more of the job's EXEC statements, satisfaction of the return code test on the JOB statement terminates the job, even if the return code test requested on the EXEC statement is not satisfied. Likewise, if the return code test requested on the EXEC statement is satisfied, the step is bypassed even if the test requested on the JOB statement is not satisfied.

Examples

```
//STEP1        EXEC   DSSAVE , COND=( 8 , LE )
```

```
//MYSTEP EXEC PRINT,COND=ONLY  
//RUNSTEP EXEC PGM=MYPGM,COND=(16,EQ)
```

PARM

Specifies variable information to be passed to the program to be executed.

```
PARM=value
```

where:

value contains up to 100 characters of information.

If the value contains any characters that are not alphanumeric or national (\$, #, @), the entire value must be enclosed in single quotes. If these special characters are commas, the value may be enclosed in parentheses instead of single quotes. The enclosing single quotes or parentheses are not passed to the program being executed.

If the value contains an ampersand or an apostrophe, two ampersands or apostrophes must be given. The program being executed will receive only one character.

If the value must be continued on another statement, it should be enclosed in parentheses and continued at a comma. The continuation comma is considered part of the value field and counts toward the maximum of 100 characters.

Examples

```
//STEPONE EXEC PGM=HEWL,PARM=MAP  
  
//S5 EXEC PGM=MYPGM1,  
// PARM=(JONES,JOHN,FEBRUARY15,1945,BETHESDA,  
// MARYLAND)  
  
//STEP26 EXEC PGM=MYPGM,PARM=('QUADRATIC,AX+BX**2')
```

REGION

Specifies the maximum amount of virtual storage to be made available to a job step "below the line."

```
REGION=nnnnK  
REGION=nM
```

where:

K equals 1024 bytes and M equals 1024K.

If the JOB statement specifies a REGION parameter, REGION parameters on EXEC statements will be ignored. It is better practice to specify the REGION parameter on the EXEC statements than on the JOB statement.

A default of 4096K is assigned to each job step when REGION is not specified (i.e., in your JCL or in the JCL of the procedure being executed).

The system allocates region in increments of 4K. Hence, a request for 201K will be interpreted as though 204K were specified.

If your job fails because of region, increase the region and rerun the job.

Examples

```
//RUNSTEP3 EXEC PGM=MYREPORT,REGION=4M
//STEP2 EXEC CBLALKGO,REGION.GO=6M
```

TIME

Specifies the maximum amount of time that a job step may use the CPU. The format is:

TIME=(time)

Replace *time* with one of the following:

mmm,ss
mmm
,ss

where

mmm represents minutes

ss represents seconds

The maximum number of minutes is 999 if no seconds are specified, 998 otherwise; the maximum number of seconds is 59. There is no default.

999 minutes is equivalent to 59,940 seconds, or 16 hours and 39 minutes. The total time for all steps may not exceed the time specified in the JOB statement.

If a step specifies TIME and requires less than the time specified, you are charged only for the time used.

A step that exceeds its allotted time will abend and cause the job to terminate.

In deciding how much time to specify, consider these points:

- Titan reports the actual time used, so after running a job once, you know how long each step should take.
- On the first execution, try to estimate an amount of time that is generous enough to allow the step to complete but short enough to stop looping.

Example

The following example executes the procedure COBOLCL to compile and link edit a COBOL program. The TIME parameter will prevent excessive costs in case a runaway condition results from a programming error.

```
//STEPY EXEC COBOLCL,TIME=( , 30)
```

2.6 DD Statement

The Data Definition (DD) statement specifies I/O devices, location of the data set, certain data attributes, and special processing options. The DD statements for a job step must follow the EXEC statement for that job step. If DD statements override statements in a procedure, they must appear in the same order as in the procedure. Any additional DD statements for a step follow overriding DD statements for that step.

Unless the program uses dynamic allocation, there must be a DD statement for each data set to be created or used in a job step. See section 2.10, *Creating and Using Data Sets*.

The DD statement consists of the characters // in columns 1 and 2, followed by four fields:

- name
- operation (the characters DD)
- parameters
- comments

The format is:

```
//ddname DD parameters comments
```

2.6.1 Restrictions at NIH

The following DD names are not supported:

SYSCHK
SYSCHKEOV

The following DD parameters are not supported:

CHKPT
DSID
QNAME
SUBSYS

2.6.2 Name Field

The name field (ddname) is optional. If supplied, the ddname consists of 1 to 8 alphanumeric or national (\$, #, @) characters and the first character must be alphabetic or national.

The ddname identifies a DD statement to which other control statements in a program can refer or identifies a DD statement which invokes a particular job control language facility. The ddname begins in column three.

The ddname should be omitted when the statement defines a data set that is concatenated to the data set of the preceding DD statement.

Each ddname within a job step should be unique. If duplicate ddnames exist in a step, all references are directed to the first occurrence in that step.

Certain ddnames are reserved for specific operating system functions. These are:

JOBLIB	Defines a private library made available to the job for searching for a program name. This specification must precede the first EXEC statement.
STEPLIB	Defines a private library made available to a job step for searching for a program name. This specification should follow the execute statement for the job step.
SYSABEND	Defines a data set on which a dump can be written if the job step terminates abnormally. The dump provided would include the processing program area, the system nucleus, and the system queue area. Since it generates large amounts of data to be printed, this statement should not be used unless requested by the NIH Help Desk.
SYSMDUMP	Defines a data set on which an unformatted, machine-readable dump can be written if the job step terminates abnormally.
SYSUDUMP	Defines a data set on which a dump can be written if a job

step terminates abnormally. The dump provided would include only the processing program area. This option is sufficient for problem determination in most cases of abnormal termination.

2.6.3 Operation Field

The operation field contains the keyword DD and must be preceded and followed by at least one blank.

2.6.4 Parameters Field

At least one parameter identifying the type of data set to be processed must be specified. Depending upon the type specified, additional parameters may also be required. The additional parameters may be in any order, separated by commas.

The following is a description of the most common data definition statement parameters.

* or DATA

Indicates that input stream data records immediately follow the DD statement. More than one input stream data set can be coded per job step if the ddnames are unique. The only other parameters that may appear on a DD statement specifying the * are DCB and DLM. The DATA parameter must be specified on the DD statement when the input records that follow contain JCL statements or data statements that have // in columns 1 and 2.

Input stream data is assumed to be 80-byte logical records. DCB information supplied by the processing program must conform to this rule. If DCB information is not supplied, the characteristics RECFM=F and LRECL=80 will be assumed. Any specification of BLKSIZE and BUFNO will be ignored.

The data records end when one of the following is found in columns 1 and 2:

- /* in the input stream
- // to indicate another JCL statement (DD * only)
- The two-character delimiter specified by a DLM parameter on this DD statement

The data records also end if the input stream runs out of card images

The following examples show valid ways to end an input stream:

```
//ddname DD *  
Input stream data records  
/*
```

```
//ddname DD DATA
Input stream data records
/*
```

```
//ddname DD DATA,DLM=$$
Input stream data records
$$
```

AMP

Provides for completing information in an access control block (VSAM data sets only).

See also the MAX utilities (section 10.6) for processing VSAM data sets.

AVGREC

For a new data set, specifies that the units of allocation for SPACE are records and that the primary and secondary space parameters represent either single records, thousands of records, or millions of records. Used with SPACE=reclength.

- U Indicates that number used for primary and secondary space allocation represents the number of records
- K Indicates that number used for primary and secondary space allocation represents that number multiplied by a thousand
- M Indicates that number used for primary and secondary space allocation represents that number multiplied by a million

CHARS

Selects up to four character sets to be used in printing a data set that is not controlled by print systems defaults or within a PAGEDDEF.

The character set parameter specifies fonts to be used when printing. The CHARS parameter is ignored if a font has been specified in a PAGEDDEF within an OUTPUT statement. If neither a CHARS parameter nor a PAGEDDEF is used, the system default fonts will be used.

COPIES

Specifies number of copies to be printed.

DATACLAS

Specifies a data class for a new data set. The data classes are names predefined by the installation and have assigned DCB and SPACE parameter values.

```
DATACLAS=data-class-name
```

where:

data-class-name is a 1-to-8 character name predefined by the installation.

The DATACLAS names for non-VSAM data sets are:

DATAF	defines a fixed block, physical sequential, 80-character data set with default primary and secondary space values of 5000 records.
DATAV	defines a variable blocked, physical sequential, 255-character data set with default primary and secondary space values of 150 records; enough space to hold at least 2,400 records (more depending on how much the average length of the records is less than the maximum length of 251).
LISTF	defines a fixed block, physical sequential, 133-character data set with ISO/ANSI carriage control and default primary and secondary space values of 350 records; enough space to hold up to 5,600 records.
LISTV	defines a variable blocked, physical sequential, 137-character data set with ISO/ANSI carriage control and default primary and secondary space values of 340 records; enough space to hold at least 5,440 records (more depending on how much the average length of the records is less than the maximum length of 133).
PDSF	defines a fixed block, 80-character partitioned data set (PDS) with a default primary space value of 250 records and secondary space value of 1,000 records. This is enough space to hold at least 15,250 records total for all members, with 20 directory blocks (sufficient for about 120 members).

Additional Information

Data classes can be used to simplify the coding of JCL by providing a default for the DCB the SPACE parameters.

- Data classes can be used when allocating data sets on dedicated disks or public disks.
- Specify the data classes in batch jobs by using the DATACLAS JCL parameter, and in TSO via the DATACLAS parameter of the ALLOCATE command. They are only applicable when a data set is being created and will

be ignored if used in reference to an existing data set. Data classes only support physical sequential (PS), partitioned (PO), and VSAM (VS) data sets.

- The non-VSAM data classes represent the most popular DSORG/RECFM combinations on the system. The VSAM data classes can be used to allocate VSAM data sets, with some limitations, directly via a batch JCL DD statement instead of executing an IDCAMS DEFINE command (using the VDSUTIL procedure).
- All space values are specified as records or bytes (instead of tracks or cylinders) making the data classes device independent, i.e., they will be valid even if the current disk storage devices (DASD) on the system are replaced with drives of a different architecture or geometry.
- The block size defaults to the optimum block size for the device on which the data set is being allocated.
- For VSAM data sets, only the cluster name of the data set is specified in the JCL. The component names are assigned by the system (usually by adding .DATA or INDEX to the end of the cluster name).
- For VSAM data classes, only the default RECORDSIZE is arbitrary and should be specified via the JCL LRECL parameter (see examples below). Likewise, the KEYS length and offset values for KSDS data sets must be specified using the KEYLEN and KEYOFF JCL parameters.
- In the VSAM data classes, the RECOVERY, NOERASE, NOREUSE and NONSPANNED default parameters cannot be overridden or modified. If there is a need to create a VSAM data set with the SPEED, ERASE, REUSE, or SPANNED attributes, then the IDCAMS DEFINE command (via the VDSUTIL cataloged procedure) must be used to create the data set.
- The following list shows the non-VSAM data classes and their equivalent DCB and SPACE parameters:

**DATACLAS
Parameter**

Equivalent DCB and SPACE Parameters

DATACLAS=DATAF	DCB=(RECFM=FB,DSORG=PS,LRECL=80), SPACE=(80,(500,500)),AVGREC=U
DATACLAS=DATAV	DCB=(RECFM=VB,DSORG=PS,LRECL=255), SPACE=(255,(150,150)),AVGREC=U
DATACLAS=LISTF	DCB=(RECFM=FBA,DSORG=PS,LRECL=133), SPACE=(133,(350,350)),AVGREC=U
DATACLAS=LISTV	DCB=(RECFM=VBA,DSORG=PS,LRECL=137), SPACE=(137,(340,340)),AVGREC=U
DATACLAS=PDSF	DCB=(RECFM=FB,DSORG=PO,LRECL=80), SPACE=(80,(250,1000,20)),AVGREC=U

Note: The above space allocations are equivalent (TRK,(1,1)) for the four PS data classes and (TRK,(1,2,20)) for the PO data class when using our current disk storage devices (3390s).

The default attributes of the four VSAM data classes are shown below:

Default Attributes for VSAM Data Classes

ATTRIBUTE	VSAM DATA CLASSES	
	VSAMKSDS	VSAMLSDS
Type	indexed	linear
Space (primary)	400KB	400KB
	Comments: Can be overridden by the SPACE parameter	
Space (secondary)	400KB	400KB
	Comments: Can be overridden by the SPACE parameter	
RECORDSIZE (avg)	256	256
	Comments: Is overridden by the LRECL parameter	
RECORDSIZE (max)	256	256
	Comments: Is overridden by the LRECL parameter	
BUFSPACE	45568	8192
	Comments: Can be overridden by the AMP parameter	
CISIZE	22528	512
	Comments: System determined optimum size for device (currently 3390). Can be influenced by RECORDSIZE (specified by the JCL LRECL parameter) and the BUFSP value (specified in the JCL DD AMP parameter)	
KEYS (length)	1	n/a
	Comments: Is overridden by the KEYLEN parameter	
KEYS (offset)	0	n/a
	Comments: Is overridden by the KEYOFF parameter	

Default Attributes for VSAM Data Classes (Continued)

ATTRIBUTE	VSAM DATA CLASSES	
	VSAMKSDS	VSAMLSDS
SHARE- OPTIONS	1,3	1,3
	Comments: Can be modified with the IDCAMS ALTER command	
recovery/speed	RECOVERY	RECOVERY
erase/noerase	NOERASE	NOERASE
writechk/ nowritechk	NOWRITECHK	NOWRITECHK
	Comments: Can be modified with the IDCAMS ALTER command	
reuse/noreuse	NOREUSE	NOREUSE
spanned/ nonspanned	NONSPANNED	NONSPANNED
FREESPACE_CI	20%	n/a
	Comments: Can be modified with the IDCAMS ALTER command	
FREESPACE_CA	20%	n/a
	Comments: Can be modified with the IDCAMS ALTER command	

NOTE: The space allocations are equivalent to (TRK,(10,10)) in terms of 3390 tracks.

Examples

The following examples illustrate how data classes can be used to allocate a new data set and how they can be modified to create a data set that does not exactly match the default values for the data class.

This example allocates a data set to hold the source for a COBOL program (up to 8000 records).

```
//DD1 DD DSN=userid.name.COBOL,DATACLAS=DATAF,  
// UNIT=DISK2YR,DISP=(NEW,CATLG)
```

This example creates a variable data set that could be used to save output from a job that has many short lines in its listing. Space may be saved by putting it into a variable file.

```
//DD1 DD DSN=userid.name.LIST,DATACLAS=LISTV,  
// UNIT=DISK2YR,DISP=(NEW,CATLG)
```

In this example, the listing has machine instead of ISO/ANSI control characters so the record format is overridden.

```
//DD1 DD DSN=userid.name.LIST,DATACLAS=LISTV,  
// RECFM=VBM,UNIT=DISK2YR,DISP=(NEW,CATLG)
```

This example has 90-character fixed length records. The LRECL parameter is used to override the default record length.

```
//DD1 DD DSN=userid.name.DATA,DATACLAS=DATAF,  
// LRECL=90,UNIT=DISK2YR,DISP=(NEW,CATLG)
```

NOTE: even though the LRECL has been increased from 80 to 90, the amount of space allocated will NOT be increased, i.e., the maximum number of 90-character records that will fit in this data set would be about 11% less than the number of 80-character records.

The following example illustrates overriding the default space values for the DATACLAS parameter.

```
//DD1 DD DSN=userid.name.DATA,DATACLAS=DATAF,  
// LRECL=200,SPACE=(200,(50,5),RLSE),AVGREC=K,  
// UNIT=DISK2YR,DISP=(NEW,CATLG)
```

The following examples illustrate the creation of VSAM data sets using the DATACLAS JCL parameter.

```
//DD1 DD DSN=userid.name.KSDS,DATACLAS=VSAMKSDS,  
// LRECL=530,KEYLEN=45,KEYOFF=10,  
// UNIT=DISK2YR,DISP=(NEW,CATLG)
```

DCB

The data control block parameter specifies data set attributes not defined in the processing program or in data set labels.

DCB Subparameters

This section discusses the commonly used BLKSIZE, LRECL, and RECFM subparameters, the infrequently used DSORG, BUFNO, and OPTCD subparameters, and the DEN subparameter for 9-track tapes.

BLKSIZE, LRECL and RECFM

DCB subparameters block size (BLKSIZE), logical record length (LRECL), and record format (RECFM) are required fields in the data control block for most applications. The format is:

```
DCB=(BLKSIZE=physical-bytes,LRECL=logical-bytes,
RECFM=record-format)
```

where:

physical-bytes	specifies the maximum number of bytes in a block (physical record). This value may not exceed 32760. For variable length records, the BLKSIZE must be 4 bytes larger than the LRECL to allow for control information.
logical-bytes	specifies the actual length, in bytes, for fixed-length records. For variable length records, the LRECL must be 4 bytes longer than the longest record to allow for control information.
record-format	specifies the format and characteristics of the data set records. The three length indicators are: F Records are of fixed length U Records are of undefined length V Records are of variable length.

Block Size - Additional Information

- Small block sizes (e.g., BLKSIZE=400) can severely degrade processing efficiency for large data sets.
- Block sizes close to but not exceeding 27998 for disk data sets and 32760 for tape are recommended.
- The simplest and most efficient thing to do is to code BLKSIZE=0 to cause the operating system to determine the optimum block size. This provides a way to use this technique even for batch cataloged procedures in which the block size is passed as a symbolic parameter on the EXEC statement.)

- When you can't code BLKSIZE=0, excluding the BLKSIZE subparameter in the DCB parameter tells the operating system to calculate the optimum BLKSIZE if LRECL is specified and RECFM is fixed or variable.
- When a data set is accessed by a batch job, that job must provide sufficient region for up to five buffers for each data set. Since each buffer must be the same size as the data set's BLKSIZE, considerable region may be required.
- Block size must be compatible with all software that is to access the data set.
- When a data set is created with the system determining the optimum block size, it will be automatically reblocked to a new optimum block size if the data set has to be moved to a new DASD device with a different track/cylinder architecture.

Record Format - Additional Information

- During a job's conversion process, testing is done to ensure that valid options are specified for the RECFM, but no verification is made for the combination of such parameters. For example, an invalid combination such as RECFM=UV will not be detected during the conversion or interpretation processes.
- Except for BDAM, supported access methods allow blocking of fixed length and variable length records. For these access methods, record-format would be one of the following:

FB	Fixed length records, blocked
VB	Variable length records, blocked

- The sequential access methods BSAM and QSAM allow S to be specified as a characteristic of fixed length or variable length records. For variable length records (RECFM=VS or RECFM=VBS), this indicates that a logical record can span more than one block. For fixed length records, (RECFM=FBS), this indicates that the records are written as standard blocks (no truncated blocks, except the last).

If a data set being read FBS has a short block in the middle (for example, if it were extended with DISP=MOD), it cannot be read properly. Attempting to read a data set as FBS specifying a block size larger than the one used to create it causes the first block to be treated as a short block.

- The BPAM, BSAM and QSAM access methods allow the RECFM to indicate that the records contain printer control characters. The letter A designates ASA control characters. The letter M designates machine control characters. The following combinations are valid:

FA	FBA	FBM
FBSA	FBSMFM	
UA	UM	VA
VBA	VBM	VBSA
VBSM	VM	VSA
VSM		

- The T designation to indicate track overflow (e.g., RECFM=FBT) is not applicable for NIH Computer Center disks, because the size of a track is greater than the maximum allowable block size.

Examples

```
//OUTRECS      DD DCB=(RECFM=FB,LRECL=100,BLKSIZE=11400),
//              UNIT=TAPE,RETPD=60,
//              DISP=(NEW,PASS),DSN=userid.JULY.REPORT

//GO.FT09F001  DD DCB=(RECFM=VBS,
//              LRECL=9000,BLKSIZE=6000),
//              UNIT=TEMP,DISP=(NEW,CATLG),
//              DSN=userid.DATA,SPACE=(TRK,(10,5),RLSE)

//STEP1 DD      DSN=userid.TESTIT,UNIT=DISK2YR,
//              DISP=(NEW,CATLG),SPACE=(TRK,(1,1,10),RLSE),
//              DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)

//OUTDATA      DD DCB=(RECFM=FB,LRECL=100,BLKSIZE=11400),
//              UNIT=NOBACKUP,DISP=(NEW,CATLG),
//              SPACE=(CYL,(10,2),RLSE),
//              DSN=userid.RECS
```

DSORG, BUFNO and OPTCD

DCB subparameters data set organization (DSORG), buffer number (BUFNO), and option code (OPTCD) are covered in this section. The format is:

```
DCB=(DSORG=organization,BUFNO=n,OPTCD=code)
```

where:

organization indicates the data set organization. If the DSORG subparameter is not provided, a DSORG of PS is assumed.

PS	Physical sequential
DA	Direct

PO Partitioned data set (PDS)

VS (for VSAM data sets) is not needed and should not be given. The use of VSAM is fully described in the following manuals: *Using Virtual Storage Access Method (VSAM) and Access Method Services at NIH, DFSMS/MVS Access Method Services (ICF)*, SC26-4906, and *DFSMS/MVS Using Data Sets*, SC26-4922.

n specifies the number of buffers to be assigned to the DCB. The defaults are 5 for QSAM and 2 for all other access methods.

code specifies optional services to be performed by the control program. Many codes are possible; four are of particular interest:

B disregard end of file (EOF) labels on tape; treats multi-volumes as a single data set.

J indicates the output contains Table Reference Characters (TRCs) to print with more than one character set.

Q translation to or from ASCII is required.

Z requests reduced error recovery for magnetic tape input; useful for problem tapes.

Refer to the appropriate language manual for information on when DSORG must be coded. For further information, see *OS/390 MVS JCL Reference*, GC28-1757.

Restrictions at NIH

- The organizations PSU, DAU, ISU and POU cannot be used at this installation because such data sets are unmoveable. This makes disk maintenance, which involves condensing and copying data sets, impossible.
- IS organization (ISAM) is not supported by the NIH Computer Center and is therefore not allowed.

Additional Information

- The number of buffers chosen affects the region size required for execution. For a small data set with a large block size, fewer buffers will require less region. For large data sets, especially those with small block sizes, the number of buffers should not be reduced because execution time will be increased.

Examples

```
//LITFILE      DD DCB=(BUFNO=2,OPTCD=Z,
//             RECFM=FB,LRECL=1000,
//             BLKSIZE=11000),
```

```

//          VOL=(PRIVATE,SER=123456),EXPDT=98000,
//          UNIT=FRGN3480,DISP=OLD,DSN=ABC1XYZ.BADTAPE

//DIRECT   DD DCB=(DSORG=DA,RECFM=F,LRECL=500,
//          BLKSIZE=500),
//          UNIT=DISK2YR,SPACE=(TRK,(10,1),RLSE),
//          DISP=(NEW,CATLG),DSN=userid.BDAM

//OUTPUT   DD SYSOUT=A,CHARS=(SN10,SE10),
//          DCB=OPTCD=J

```

DEN for 9-Track Tapes

The DEN subparameter specifies the density of a tape. DEN is used only with 9-track tapes; it does not apply to cartridge tapes. DEN should be coded for all 9-track tapes. The format is:

```
DCB=(DEN=density)
```

where:

density a number which indicates the number of bits-per-inch (BPI) used to write a data set.

DEN=3 indicates 1600 BPI (9-track)

DEN=4 indicates 6250 BPI (9-track)

The NIH tape drives ensure that any data set added to a tape is written with the same density and parity as the existing data. Any specification of other values will be ignored.

For a complete description of the tape attributes supported by the NIH Computer Center see the *Titan User's Guide*.

Example

```

//GO.INPUT   DD DCB=(RECFM=FB,LRECL=100,
//          BLKSIZE=6200,DEN=4),EXPDT=98000,
//          UNIT=FRGN6250,VOL=(PRIVATE,SER=012345),
//          LABEL=(1,NL),DISP=(OLD,KEEP),
//          DSN=userid.IN1

```

DDNAME

Postpones definition of a data set until a DD statement later in the same job step.

DEST

Specifies a destination for output. The destination can be LOCAL, a remote workstation, a node, or a workstation at a node. Use the format:

DEST=dest

For this destination	Use this value for dest
central printers	LOCAL
remote workstation nnnn	Rnnnn or RMnnnn or RMTnnnn
NJE node connected to NIH	nodename
a remote workstation connected to an NJE node	nodename.remote

DISP

Describes the status of the data set at the beginning of the job step, the disposition of the data set if the job step terminates normally, and the disposition of the data set if the job step terminates abnormally.

Specify DISP=SHR for all existing public disk data sets and for all other disk data sets involved in the job that are not being updated. Specify DISP=(NEW,CATLG) for all new public data sets or generation data sets.

DLM

Allows specification of a delimiter other than /* to terminate an input stream data set.

DSNAME

Assigns a name to a data set. This parameter is not necessary if the data set is temporary (i.e., if the data set is being created and deleted within the job).

Note: Titan requires that permanent data set names on tape or disk begin with either a userid (userid.dsname) or an account group (account.dsname).

In addition, user-specified data set names can not exceed 44 characters, including required characters, optional characters, and all periods. For each group of eight characters or fewer there must be a period, and the first character following each period must be alphabetic or national (@, #, or \$); the remaining characters can be alphabetic, numeric, national or hyphen. Some data set naming conventions will assume specific management classes and take precedence over the unit name specified:

Disk data set names ending with the final qualifier OUTLIST or SYSOUT are forced to be UNIT=TEMP.

Disk data set names ending with the final qualifier ASM, CLIST, CNTL, COB*, EXEC, FOR*, JCL, LIB, *LIB, LLIB, LOAD*, OBJ, PGM*, PL1, PLI, RESLIB, SCRIPT, SOURCE, SRC, or TEXT are forced to be UNIT=LONGTERM (where * is a wildcard representing zero or more characters).

Temporary data set names must begin with two ampersands (&&) followed by a character string 1 to 8 characters long. The first character following the ampersands must be alphabetic or an @, #, or \$.

Examples of DSNNAME

The following examples show the format of a data set name for a file created by a Titan user.

DSNNAME=userid.dsname

or

DSNNAME=account.dsname

or

DSNNAME=&&dsname

DUMMY

Specifies that all I/O operations are to be ignored for this data set.

EXPDT

Use the EXPDT parameter to specify the expiration date for a new tape data set. For more information, see Section 11.2. The available EXPDT keywords and their meanings are:

EXPDT=yyyy/ddd Specifies the date when a tape data set can be deleted. Example: EXPDT=2003/180 for the 180th day of the year 2003.

EXPDT=98000 Specifies a foreign tape to the Tape Management System (TMS), and is required to read or write a tape which is not under the control of TMS.

EXPDT=99000 Specifies a tape data set that is to be scratched if and when it becomes uncataloged.

EXPDT=99365 Specifies a tape data set that is to be kept

indefinitely. Data sets with this expiration date cannot be recreated on the same tape; however, these data sets may be added to using DISP=MOD.

FCB

Vertical spacing and page ejects are controlled by a Forms Control Buffer (FCB) parameter when printing is directed to the IBM 4245 impact printer. The only FCB's supported are those used with Computer Center-supplied labels or registered with the Computer Center. For available FCBs, see section 2.12.

FREE

Specifies whether a data set referred to by the FREE parameter will be deallocated when either the data set is closed or the job step ends. This parameter should be used to prevent monopolizing critical resources. The FREE parameter may be used with all SYSOUT data sets.

The FREE parameter has two subparameters:

CLOSE FREE=CLOSE will cause the data set to be deallocated as soon as it is closed. FREE=CLOSE should not be included in a DD statement for a data set that is opened and closed more than once during a job step. FREE=CLOSE is ignored for a DD statement that specifies JOBLIB, STEPLIB, or a data set of a concatenated group.

END FREE=END will cause the data set to be deallocated at the end of the job step.

HOLD

HOLD=YES places a SYSOUT data set on the HOLD queue until released by the central operator or by a remote terminal operator or user. All jobs in HOLD are purged (except for SYSOUT CLASS=I which is printed) after 7 calendar days—this includes weekends. If for some reason a job is awaiting output for more than 14 calendar days, it will be purged.

LABEL

Defines the type of label processing to be performed for a data set. The format is:

`LABEL=(seqno,type,password,process,retention)`

where:

seqno is the relative position of a data set on tape. If this subparameter is not specified, the system will supply a default value of 1.

type	specifies the label type. There are five possible label types:
SL	IBM standard labels. All Titan library tapes have IBM standard labels. If a label type is not specified, the system will default to SL.
AL	American National Standards Institute (ANSI) labels.
NL	No labels.
NSL	Not supported.
BLP	Bypass label processing. This option is not supported. (See section 11.1.)
password	not supported.
process	specifies the type of processing for which this data set is being allocated:
IN	The data set is to be used for input only. All FORTRAN users must specify IN if the data set is to be used for input only; failure to do so will cause operator intervention and cancellation of the job.
OUT	The data set is to be used for output only.
INOUT	The data set is to be used for both input and output.
retention	supplies information about the retention period or expiration date of a tape data set. For more information, see Section 11.2. IBM recommends that you specify the EXPDT DD statement instead of the retention parameter. The available keywords are:
EXPDT=yyyy/ddd	Specifies the date when a tape data set can be deleted. Example: EXPDT=2003/180 for the 180 th day of the year 2003.
EXPDT=98000	Specifies a foreign tape to the Tape Management System (TMS), and is required to read or write a tape which is not under the control of TMS.

EXPDT=99000	Specifies a tape data set that is to be scratched if and when it becomes uncataloged.
EXPDT=99365	Specifies a tape data set that is to be kept indefinitely. Data sets with this expiration date cannot be recreated on the same tape; however, these data sets may be added to using DISP=MOD.
RETPD=nnn	Specifies the number of days that a tape data set must be retained. Leading zeroes are not required.

Examples:

The following example indicates the data set is to be used for input only. Any attempt to overwrite or delete data from this data set will be denied.

```
LABEL=( , , , IN)
```

The following example of the SL option indicates the data set is the second data set on the volume and has a standard IBM label.

```
LABEL=( 2 , SL )
```

The following example shows the format for retaining a data set 30 days from the date of creation.

```
LABEL=RETPD=30
```

MGMTCLAS

Specifies the name of a management class to be used to manage the data set after the data set is allocated. The management class determines how long the data set will be kept before it is automatically scratched and whether or not it will be backed up. See the DSNAMES parameter on page 47 for data set name qualifiers that cause the system to assign a management class and ignore the specified MGMTCLAS.

If MGMTCLAS is specified, UNIT should not be.

TEMP	Disk data set on public volume with TEMP management class that is automatically scratched after 7 days. No backups.
NOBACKUP	Disk data set on public volume with NOBACKUP management class that is not automatically scratched. No backups.

DISK2YR	Disk data set on public volume with DISK2YR management class that is automatically scratched 2 years after last use. Up to 5 incremental backups.
DISK7YR	Disk data set on public volume with DISK7YR management class that is automatically scratched 7 years after last use. Up to 5 incremental backups.
LONGTERM	Disk data set on public volume with LONGTERM management class that is not automatically scratched. Up to 5 incremental backups.

OUTLIM

Allows specification of a limit for the number of logical records included in a SYSOUT data set.

OUTPUT

Associates a SYSOUT data set with one or more OUTPUT JCL statements. The format is:

```
OUTPUT=* .name
```

where "name" is the name of the OUTPUT JCL statement.

For example:

```
//SAMP1 OUTPUT COPIES=2
//MINE DD SYSOUT=A,OUTPUT=* .SAMP1
```

Enclose multiple references to OUTPUT JCL statements in parentheses and separate the references with commas. For example:

```
//OUT1 OUTPUT DEST=STLNODE.RMT1234
//OUT2 OUTPUT COPIES=2
//DS DD SYSOUT=A,OUTPUT=( *.OUT1 , *.OUT2 )
```

RETPD

Specifies the number of days that a new tape data set must be retained. Leading zeroes are not required. For more information, see Section 11.2.

SPACE

Requests space on a disk volume.

```
SPACE=(TRK,(primary,secondary,directory),RLSE,CONTIG)
      CYL
      blocklength
```

reclength

where:

TRK	specifies that space is to be allocated in units of tracks.
CYL	specifies that space is to be allocated in units of cylinders.
blocklength	specifies the average block length of the data in bytes. The system computes the number of tracks to allocate. NOT used if AVGREC is coded.
reclength	specifies the average record length of the data in bytes. The system computes the block size and how many tracks to allocate. Used only if AVGREC is coded.
primary	specifies the number of tracks, cylinders, or blocks required to store the data set. When the AVGREC parameter is coded, primary represents the number of records. For a PDS, the number of records is the total number of records in all members.
secondary	specifies the number of tracks, cylinders, or blocks to be allocated if additional space is required. This additional space should reflect the expected growth of the data set. (Five to ten percent of the <i>primary</i> value is a safe value.)
directory	specifies the number of 256-byte directory blocks to be contained in the directory of a partitioned data set. Each directory block has room for approximately 4-7 member entries.
RLSE	indicates that space not used by the data set is to be released; RLSE should be specified whenever possible. If you do not specify secondary, RLSE is ignored.
CONTIG	specifies that the primary space allocated must be contiguous.

If *secondary* and *directory* are omitted, the parentheses around *primary* are not needed. A comma must be coded if *secondary* is omitted and *directory* is coded. Similarly, a comma must be coded if RLSE is omitted and CONTIG is coded.

Restrictions at NIH

The ABSTR subparameter is not supported.

Additional Information

- The SPACE parameter is ignored for tapes and existing disk data sets.

-
- The SPACE parameter must be provided when any data set is created on a disk.
 - The NIH Computer Center recommends that you estimate the amount of space you need and allocate this space in primary and use the RLSE option. Secondary space should be about 10 percent of the primary space allocation. If no secondary space is specified, unused space is not released.
 - Space on the public volumes can be allocated in tracks, cylinders, blocks, or average record length of the data in bytes.
 - If the primary allocation cannot be satisfied on a single volume in 5 extents (noncontiguous areas), the job will ABEND.
 - If a secondary quantity is required, there must be enough space in up to 5 extents (noncontiguous areas) to satisfy each request for the secondary quantity. Additionally, no more than 16 extents will be used to satisfy the entire SPACE request (primary and total secondary). If this space is not available, the job will ABEND.
 - Expansions of existing data sets use the secondary quantity to allocate more space. If you anticipate expansion or if you think your primary space allocation is insufficient, use the secondary allocation.
 - Directory space must be allocated when a partitioned data set is created. A DSORG parameter of PO must be specified. See section 2.6.4 for additional information.
 - If CONTIG is specified and the requested contiguous space is not available, the job will ABEND. Contiguous space should not be requested unless it is required by the program. Requesting it reduces the chance of there being suitable space available. Contiguous space will always be allocated if it is available.
 - If contiguous space is requested and a secondary quantity is coded, the primary request will be satisfied with contiguous space, but the secondary quantity may not be contiguous.
 - On public disk volumes, a single data set may not be split between volumes (i.e., multi-volume data sets).

Restrictions for BDAM Data Sets

- BDAM data sets cannot be expanded once they are created.
- For relative block address BDAM data sets, it is more efficient and cost-effective to request only the number of tracks needed. Writing dummy records to fill unneeded space can increase I/O, CPU, and storage costs.

- Key Length must be considered for some BDAM data sets when calculating space requirements on a 3390.

SYSOUT

Assigns an output class to a data set or sends the data set to the internal reader as an input job stream. The format is:

```
SYSOUT=(class,program,form)
```

This example sends the data set to the internal reader:

```
SYSOUT=( ,INTRDR)
```

Subparameters of SYSOUT

The subparameters are positional.

class	Required if not using the internal reader. Assigns the output to an output class.
program	Optional. Not used, but if the <i>form</i> subparameter is used, this missing parameter must be represented by an extra comma.
form	Optional. Specifies a form type to be used for this output data set. If this subparameter is not specified, the default is STD. For information about forms, see Printing in section 2.12.
INTRDR	Tells JES that this sysout data set is to be sent as an input job stream to the internal reader, which reads and interprets it.

The output classes are listed below:

SYSOUT Classes

Class	Attributes
A	Output will be printed.
B	Reserved for internal use.
H	Output is in HOLD for 7 days to allow it to be viewed and disposed of by user. If it is still in HOLD after 7 days, it is purged.
I	Output is in HOLD for 7 days to allow it to be viewed and disposed of by user. If it is still in HOLD after 7 days, it is printed.
J	Junk—output is immediately discarded by JES2 without printing.
M	Reserved for SMTP output.
T	Same as class H.
*	SYSOUT class is the same as the MSGCLASS parameter on the JOB statement or the system default (A) if no MSGCLASS coded.

If an output class other than one documented above is specified and the job is run at the NIH, the output is discarded immediately. If the job is run at another node and output is sent to the NIH, it is discarded within an hour or two.

UNIT

Specifies the type of input or output device required to process the data set. See the DSNNAME parameter for data set name qualifiers that cause a management class assignment and override a specified UNIT name.

Unit names TEMP, NOBACKUP, DISK2YR, DISK7YR, and LONGTERM can be used as values for MGMTCLAS instead of UNIT. If MGMTCLAS is specified, UNIT should not be.

The following unit names are used:

TEMP	Disk data set on public volume with TEMP management class that is automatically scratched after 7 days. No backups.
NOBACKUP	Disk data set on public volume with NOBACKUP management class that is not automatically scratched. No backups.
DISK2YR	Disk data set on public volume with DISK2YR management class that is automatically scratched 2 years after last use. Up to 5 incremental backups.
DISK7YR	Disk data set on public volume with DISK7YR management class that is automatically scratched 7 years after last use. Up to 5 incremental backups.
LONGTERM	Disk data set on public volume with LONGTERM management class that is not automatically scratched. Up to 5 incremental backups.
SYSDA	Scratch disk space (for temporary data sets only).
3390	Disk data set on a non-public, non-SMS managed volume (dedicated disk). This unit name should only be used when it is necessary to allocate on a specific volume.
TAPE, CTAPE	Standard 3490 tape cartridges under control of TMS. These tapes actually reside in the Virtual Storage Manager (VSM).
FRGN3480	Foreign 3480 tape cartridges not under the control of TMS.

FRGN3490	Foreign 3490 tape cartridges not under the control of TMS.
FRGN6250, 9TRACKHI	Foreign 9-track 6250/1600 bpi reel tapes not under the control of TMS.

Default or Absent UNIT Specification

The default management class for data sets created through TSO will be DISK2YR, with the TSO UNITNAME command available to easily change the default. This command can be automatically executed whenever the user logs on by including it on the Command line in the logon screen or in a CLIST that is executed from that line.

The unit information specified on the JCL statement will override unit information that is available from other sources (i.e., the system catalog or cataloged procedures). Many of Titan's cataloged procedures contain symbolic parameters that specify a default value for the UNIT parameter. If the cataloged procedure's default is not applicable, it must be overridden or nullified.

A typical link edit cataloged procedure specifies on the SYSLMOD output DD statement a temporary data set named DSNAME=&&GOSET(GO) on a direct access scratch space UNIT=SYSDA. To use a private, cataloged data set, the user's overriding output DD statement should be coded:

```
//LKED.SYSLMOD DD DSNAME=userid.LOAD(TEST),DISP=SHR,UNIT=
```

By not specifying anything after the equal sign in the UNIT parameter, the default parameter UNIT=SYSDA in the cataloged procedure is nullified. The system catalog will be searched for data set userid.LOAD to find the correct unit and volume specifications.

VOLUME

Specifies the volume serial number (tape or disk) on which a data set resides or will reside.

2.7 OUTPUT JCL Statement

The OUTPUT JCL statement is used to:

- Specify the characteristics of the processing options for a specific SYSOUT data set
- Group SYSOUT data sets for processing by a printer
- Specify default options for output data sets
- Specify a destination for an output data set

Note: IBM recommends that you use the OUTPUT JCL statement instead of the JES2 OUTPUT statement because the OUTPUT JCL statement has enhanced output processing capabilities.

The OUTPUT JCL statement consists of the characters // in columns 1 and 2, followed by four fields:

- Name
- Operation (the word OUTPUT)
- Parameters
- Comments

The format is:

```
//name OUTPUT parameters comments
```

The OUTPUT JCL statement must precede any SYSOUT DD statement that refers to it. A SYSOUT DD statement can refer to an OUTPUT JCL statement in two ways:

- Explicitly. The SYSOUT DD statement contains an OUTPUT parameter that specifies the name of the OUTPUT JCL statement.
- Implicitly. The SYSOUT DD statement does not contain an OUTPUT parameter. An implicit reference points to one or more default OUTPUT JCL statements. The job should have at least one default OUTPUT JCL statement. It can have more than one. See the DEFAULT=YES parameter in section 2.7.4.

2.7.1 Restrictions at NIH

The following parameters are not available:

CKPTLINE	INDEX	WRITER
CKPTPAGE	LINDEX	COMPACT
CKPTSEC	PRMODE	THRESHLD
PRTY	FLASH	

2.7.2 Name Field

The name is 1 to 8 alphanumeric or national (\$, #, @) characters and the first character must be alphabetic or national.

The name field is required and must be unique within the job.

2.7.3 Operation Field

The operation field contains the keyword OUTPUT and must be preceded and followed by at least one blank.

2.7.4 Parameters Field

The following parameters are available. See the Printing section (2.12) for details of available character sets, forms, form definitions, page definitions, and other specific output attributes.

CHARS=chars

Specifies up to four character sets to be used in printing a data set when not controlled by a PAGEDEF.

CLASS=class

Specifies the sysout class for data set. For a list of classes, see the SYSOUT classes in the parameters on the DD statement (section 2.6.4).

COPIES=nnn

Specifies the number of copies to be printed. Numeric value from 1 to 255.

DEFAULT=YES

Specifies that the OUTPUT JCL statement can be referenced implicitly. An OUTPUT JCL statement containing the DEFAULT=YES parameter is called a default OUTPUT JCL statement.

DEST=dest

Specifies the destination of the output. See the DEST parameter of the DD statement in section 2.6.4.

FCB=fcb

Specifies the four-character forms control buffer for vertical spacing. Each form has a default FCB. For available FCBs, see Printing in section 2.12.

FORMDEF=fdef

Specifies the form definition, which indicates the layout of the output on a laser printer. Can also specify an overlay name created with the Overlay Language (OGL). For available form definitions, see Printing in section 2.12.

FORMS=form

Specifies form to be used for printed output. For available forms, see Printing in section 2.12.

JESDS=ALL

Requests that all system generated data sets for the job be processed according to the parameters on this OUTPUT JCL statement.

PAGEDEF=pdef

Specifies a number that indicates the lines per inch, number of lines on a page, print direction, and font selection for laser printers. For available page definitions, see Printing in section 2.12.

ROOM=box

Specifies an output box on central campus. To specify a Parklawn box, place the letter P before the box number. For more information about box numbers, see section 2.12.2.

TITLE=title

Identifies a report title to be printed on separator pages. Replace *title* with 1-60 characters. Place it in single quotes if it contains spaces or special characters.

2.7.5 Example of OUTPUT JCL Statement

OUTPUT JCL statements are useful in processing the output of one SYSOUT data set in several ways. For example, a SYSOUT data set can be sent to a distant site for printing, as shown in statement OUT1, while two copies of it are also printed locally, as shown in statement OUT2:

```
//OUT1 OUTPUT DEST=STLNODE.RMT1234
//OUT2 OUTPUT COPIES=2
//DS DD SYSOUT=A,OUTPUT=( *.OUT1, *.OUT2 )
```

2.8 COMMENT Statement

The comment statement consists of the characters `/*` in columns 1-3 followed by a comment. The comment can be typed anywhere in columns 4-80. The format is:

```
/* comment
```

A comment statement may be inserted anywhere after the JOB statement, including between the sections of a continued statement.

Examples

```
/*DATA MUST FOLLOW FORMAT 1

/* *** CALL TO INDEX ROUTINE ***

/* ==> EXECUTED FROM DSN $BEC.MYDATA <==
```

2.9 Standards for Batch Jobs

This section defines the maximum limits for jobs running on Titan and specifies those JCL parameters that may not be used.

2.9.1 Maximum Limits for Batch Jobs

The following overview defines maximum limits for batch jobs.

Maximum Limits for Batch Jobs

LIMITS FOR A SINGLE JOB	
Number of steps	255
Number of instream procedures	15
Region (below the line)	8M
Lines output	4 million
Scratch disk space allocated at one time (tracks)	100,000
CPU time	999 minutes
Region (above the line)	2047M
Total tape drives in use at one time	7
Reel tape drives in use at one time	2
LIMITS ON NUMBER OF JOBS FROM ONE USER	
Total jobs in the system	150
Jobs AWAITING EXECUTION and EXECUTING	50

In addition to the *below the line* region requested by the REGION parameter (below the 16-megabyte line in virtual storage), each job is automatically assigned at least 32M (32,768K) of requestable virtual storage above the 16-megabyte line (requiring 31-bit addressing for use).

The limits on numbers of jobs apply to jobs submitted by a single user. These limits are necessary to prevent overloading the operating system queues and to allow timely processing of jobs for all users. The limit on total jobs includes jobs in all categories (waiting to execute, executing, waiting to print, in output hold, etc.). If for some reason a job is awaiting output for more than 14 calendar days, it will be purged.

The maximum amount of public disk scratch space that can be allocated to a job depends on the requirements of the jobs in the system at the time. If your job fails because of lack of available scratch space, try the following options:

- If you can, reduce the amount of requested space.
- If you are already requesting only the required space, run your job again later.
- If neither of the above options solve the problem, consult with the NIH Help Desk.

2.9.2 Restrictions at NIH

CIT does not support certain JCL parameters on Titan; for example, those that may require operator responses or those related to functions, products, and hardware that are not installed on the system. Additionally, CIT does not support JES2 functions requiring special data sets (e.g., checkpoint/restart).

The following JCL statements have parameters should not be used at this facility:

- JOB statement (see section 2.2.1)
- EXEC statement(see section 2.5.1)
- DD statement(see section 2.6.1)
- OUTPUT JCL statement(see section 2.7.1)

In addition, do not use:

- JES3 parameters not supported by JES2
- JES3 statements
- JES2 command statements
- JCL COMMAND statements
- COMMAND statements

For more detailed information on job control language and IBM utilities, refer to the IBM documentation.

Box Numbers

When you are initially registered on Titan, the box number field is set to NOBX or NONE unless your account sponsor has provided a box number. To see if a box number is associated with your userid, select option L on the CIT/Titan Primary Option Menu. Then select option 1. If your box number is displayed in the Mailbox field, it is associated with your userid and all jobs you run.

If you try to run a job on Titan with a box number of NOBX, the job will fail with a JCL error. If you plan to use the central printers, ask your account sponsor to assign you a box number. If you do not plan to use the central printers, you should set the box number to NONE.

To change the box number associated with your userid, access the Titan Customer Locator web page at <http://titan.nih.gov/locator>. You can also override your default box number by specifying a box number in your JCL. Use the ROOM parameter of the OUTPUT JCL or /*JOBPARM statement or use the /*BOX statement.

For more information about printing, see section 2.12.2.

2.9.3 Job Turnaround

When a job has completed execution, it enters a queue to be printed or it is placed in OUTPUT HOLD (from which the user can selectively examine the output at a workstation). Use Interactive Output Facility (IOF) to track the progress of a job and to look at the job output.

Output turnaround is from the time a job enters the print queue until the output has been placed in the user's output box. Time can vary widely depending on the services chosen. For jobs printed at the central facility with the standard printing requirements, the time for the job to be printed, separated from other jobs, and routed to the appropriate output box will normally be two hours or less during normal working hours. The time required for the output to be placed in the box will vary since courier times may be involved (e.g., for output directed to the boxes located at the Parklawn site). For information on the courier service, see the *Titan User's Guide*.

2.10 Creating and Using Data Sets

Creating and accessing data sets is a primary concern for any computer application. Two general classifications of data sets are permanent and temporary. Permanent data sets can exist before and after a batch job executes. Temporary data sets can exist only for the duration of a single batch job. This section provides the requirements and restrictions for creating permanent and temporary data sets on tape and disk. The section also describes the techniques for referencing them in a job: via the system catalog, passed from a previous step, or directly (specifying all data set storage information).

See section 2.6.4, *DD Statement*, for an explanation of each parameter.

2.10.1 Creating Temporary Data Sets on System Scratch Disks

Temporary data sets can be created and used on system scratch disks. Using scratch disks for temporary data storage is generally more efficient and less prone to problems arising from operator and hardware errors than using scratch tapes.

Required DD Parameters

- UNIT=SYSDA specifies that a system scratch disk is to be used.
- SPACE specifies the space to be allocated for the data set. See the Titan User's Guide for details on estimating space for any new data set. Note that key length (KL) must be considered for some BDAM data sets when calculating space requirements on a 3390.

Optional DD Parameters

- DISP=(NEW,PASS) is needed if the data set is to be used in a later step.

- DSNNAME can be used for ease of reference if a data set is passed; if no DSNNAME is given, the system assigns a unique name. If a DSNNAME is specified, it must be a temporary data set name beginning with two ampersands (&&) followed by 1-8 characters.
- DCB is needed if the desired DCB characteristics for the data set are not supplied by the program.

Restrictions at NIH

- If a job exceeds the maximum of 100,000 tracks, it may be cancelled.
- It may not be possible to allocate the maximum of system scratch space to a single data set. The primary space must fit on a single disk. If the unit count field of the UNIT parameter on the DD statement is not coded, then all the secondary space must also be acquired on the same disk. If a single sequential or partitioned data set requires a large space allocation, coding UNIT=(SYSDA,2) on the DD statement will increase the chances of finding the needed space.

Examples

```
//SORTWK01   DD UNIT=SYSDA,SPACE=(CYL,(10,1),RLSE)

//TEMPOUT    DD UNIT=SYSDA,SPACE=(11440,(310,10),RLSE),
//           DISP=(NEW,PASS),DSNAME=&&OUT,
//           DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
```

2.10.2 Creating and Deleting Data Sets

The system catalog associates the data set name with the volume serial number and unit type of the volume on which it is stored. Data sets are entered into and deleted from the catalog by appropriate specification in the DISP parameter. All disk data sets on public volumes must be cataloged.

Required DD Parameters

- DSNNAME must use the format userid.dsname or account.dsname.

Optional DD Parameters

- UNIT specifies the type of device to be used. Required when the data set is created.
- VOLUME specifies the volume information for the cataloged data set. Required only if the data set is being written on a dedicated disk.
- LABEL must be specified for tapes if the data set is not the first data set on the tape.
- SPACE must be specified when creating disk data sets.

- DCB should be specified if the desired DCB characteristics for the data set are not supplied by the program or by the data set label.

Additional Information

- Only data sets (including tape data sets) having standard data set names can be cataloged. A standard data set name consists of two or more groups of one-to-eight characters, separated by a period. (The first group is called the high-level index.) The first character of each group must be alphabetic or national (#,@,\$); the remaining characters can be alphabetic, numeric, national or hyphen. The data set name, including the periods, must not exceed 44 characters in length.
- Apostrophes are required for the data set names of non-standard data sets read from foreign tapes. Non-standard data set names cannot be cataloged.

```
//MYTAPE      DD DSNAME='THISNAMEIS.NONSTANDARD' ,
//              UNIT=FRGN3490 ,DISP=( OLD ,KEEP ) ,
//              VOLUME=SER=123456 ,EXPDT=98000
```

- If a cataloged data set is deleted (e.g., DISP=(OLD,DELETE)), but the DD statement specified the volume and unit for the data set, the data set will be deleted but not uncataloged. In this case, use the DSSCR procedure or ISPF panel 3.2 or 3.4 to uncatalog the data set.

Examples

```
//WORKFILE   DD DSNAME=userid.WORKDATA,UNIT=DISK2YR ,
//              DISP=(NEW,CATLG,DELETE) ,
//              SPACE=(TRK,(20,1),RLSE)

//STATFILE   DD DSNAME=userid.STATS,UNIT=NOBACKUP ,
//              DISP=(NEW,CATLG,DELETE) ,
//              SPACE=(CYL,(5,1),RLSE) ,
//              DCB=(RECFM=FB,LRECL=150,BLKSIZE=0)

//S1         EXEC PGM=IEFBR14
//ALLOCATE   DD DSNAME=account.filename ,
//              DISP=(NEW,CATLG),UNIT=TEMP ,
//              SPACE=(CYL,(1,1),RLSE)
```

2.10.3 Creating Permanent or Temporary Data Sets on Tape

Although temporary data sets may be created and used on system scratch tapes, they should be avoided unless the user has determined that they are required (e.g., because the permissible public scratch disk space has been fully used). Using scratch disks for temporary data storage is generally more efficient and less prone to problems arising from operator and hardware errors.

Required DD Parameters

- UNIT specifies the type of tape drive to be used for the data set. (See the UNIT parameter in section 2.6.4 for the valid device names.)

Optional DD Parameters

- DSNNAME must be specified if the data set is a permanent data set. If the data set is temporary, DSNNAME may be specified for ease of reference.
- DISP must be specified if the default of DISP=(NEW,DELETE) is not desired. DISP=(NEW,PASS) can be specified for either a temporary or permanent data set. DISP=(NEW,CATLG) can be specified only for permanent data sets.
- VOLUME must be coded if a specific tape volume is desired; otherwise, a scratch tape is used and assigned to the userid or account group specified by the high-level qualifier of the first data set on the tape.
- LABEL should be supplied if the data set is not the first data set on the tape, if the tape is not an IBM standard labeled (SL) tape, or if the data set is to be read by a FORTRAN program. NIH recommends that no more than 100 data sets should be put on one tape.
- DCB should be specified if the desired DCB characteristics for the data set are not supplied by the program.
- Either the retention period (RETPD=) or expiration date (EXPDT=) must be specified for permanent data sets (see Section 11.2). Temporary data sets on tape are released during the night following their creation regardless of the retention period or expiration date specified.

Examples

```
//SORTOUT      DD UNIT=TAPE,DSNAME=&&XYZSRT,
//              DISP=(NEW,PASS)

//GO.FT08F001  DD UNIT=TAPE,DSNAME=userid.SUMS,
//              LABEL=(2,,IN),DISP=OLD,RETPD=5,
//              VOLUME=(PRIVATE,SER=123456),
//              DCB=(RECFM=FB,LRECL=150,BLKSIZE=11400)
```

2.10.4 Using Permanent Data Sets

Permanent data sets can be stored on disk or tape. For information on the standards for each type of storage, see the *Titan User's Guide*.

Required DD Parameters

- DSNNAME specifies the data set name.
- DISP specifies the status of the data set both before and after it is referenced. The first DISP subparameter must be either OLD, MOD, or SHR.

Optional DD Parameters

- UNIT specifies the type of device assigned to a data set. Use for uncataloged data sets. Not recommended for cataloged data sets.
- VOLUME specifies the volume the data set is stored on. Use for uncataloged data sets. Not recommended for cataloged data sets.
- LABEL should be supplied if the data set is not the first data set on the tape, if the tape is not an IBM standard labeled (SL) tape, or if the data set is to be read by a FORTRAN program. NIH recommends that no more than 100 data sets should be put on one tape.
- DCB should be specified if the desired DCB characteristics for the data set are not supplied by the program or by a data set label. The DEN subparameter should be coded only for 9-track tapes.

Examples

```
//TOTALS      DD DSNNAME=userid.SUMS,UNIT=TAPE,
//              DISP=(OLD,KEEP),VOLUME=(PRIVATE,SER=123456),
//              LABEL=2

//SORTIN      DD DSNNAME=userid.INDAT,DISP=SHR

//OUTPUT      DD DSNNAME=userid.DATA,UNIT=DISK2YR,
//              SPACE=(TRK,(15,2),RLSE),DISP=(NEW,CATLG)
```

2.10.5 Using a Passed Data Set

A passed data set is one which is allocated to a job step and is not deallocated at step termination. It remains available to subsequent steps of the same job.

Required DD Parameters

- DSNNAME specifies the data set name of the passed data set.
- DISP specifies the status of the data set both before and after it is referenced. The first DISP subparameter must be OLD, MOD, or SHR.

Optional DD Parameters

- LABEL should be specified for tapes if the data set is not the first data set on the tape or if the tape is not an IBM standard label (SL) tape.

- DCB should be specified if the desired DCB characteristics for the data set are not supplied by the program or by the data set label.

Example

```
//TEMPOUT DD DSN=&&XYZOUT,DISP=(NEW,PASS),
//          SPACE=(TRK,(500,100),RLSE),UNIT=SYSDA,
//          DCB=(LRECL=133,BLKSIZE=0,RECFM=FBA)
//stepname EXEC PGM=....
//TEMPIN DD DSN=&&XYZOUT,DISP=(OLD,DELETE)
```

2.11 Using Cataloged Procedures

Production jobs that require many job control language statements can be simplified by using cataloged procedures. A cataloged procedure is a set of job control language statements that has been assigned a name and is placed in a partitioned data set (PDS). The PDS can be in a public procedure library or in a private procedure library. The private procedure library is referenced by including a JCLLIB statement in the job (see section 2.4). The cataloged procedure that contains the JCL statements is retrieved by giving its name on an EXEC statement in the input stream. Additions and modifications can be made with override statements (described in section 2.11.6). For a more complete explanation of the uses and restrictions of cataloged procedures, see the manual *OS/390 MVS JCL Reference*, GC28-1757.

A procedure contains one or more procedure steps; each step consists of an EXEC statement that identifies the program to be executed and DD statements that define the data sets to be used or produced by the program. The program requested on the EXEC statement must exist in the system library or in a private library defined by a STEPLIB DD statement.

A cataloged procedure must not contain JOB statements, delimiter statements, null statements, JOBLIB DD statements, or DD statements with * or DATA coded in the parameter field.

WARNING: Users should avoid copying JCL from NIH Computer Center procedures whenever possible. The internals of procedures may be changed at any time without warning as long as there is no change in the way they appear to the user. Copied JCL may thus become unusable.

2.11.1 JCLLIB Statement

Cataloged procedures are stored in partitioned data sets (PDSs) known as procedure libraries or JCL libraries. The JCLLIB statement is used in a job to specify libraries containing procedures and INCLUDE groups to be used in JCL. The libraries that JCLLIB points to are searched before the system libraries. (See section 2.11.8 for information on INCLUDE groups.) The format of the JCLLIB statement is:

```
//[name] JCLLIB ORDER=(userid.library1,userid.library2,...)
```

where *name* is optional and follows the rules for the ddname field in a DD statement. The JCLLIB statement must appear after the JOB statement and before the first EXEC statement. Each procedure library specified must be cataloged. The libraries are searched before any system procedure libraries and are searched in the order specified. For example,

```
//MYPROCS JCLLIB ORDER=(userid.MYLIB1,userid.YOURLIB2)
```

specifies that the procedure libraries userid.MYLIB1 and userid.YOURLIB2 will be searched for procedures in the JCL that follows it. If only one library is specified, the parentheses are optional.

2.11.2 Defining Symbolic Parameters in a Cataloged Procedure

Symbolic parameters allow a programmer using a cataloged procedure to modify the procedure easily. When the procedure is called, the EXEC statement is used to assign values to the symbolic parameters.

A symbolic parameter consists of one-to-seven alphanumeric or national (#, @, \$) characters preceded by a single ampersand. The first character must be alphabetic or national. In the cataloged procedure, specification of symbolic parameters for data set names, volume serial numbers, etc., permits the creation of a general procedure. In this manner one procedure may suffice where several might otherwise be required.

In the following example of a cataloged procedure, the data set names and tape serial numbers for the input and output data sets and the region size are left as variables; i.e., they have no default values.

Examples

```
//UPDATE PROC REGSIZE= ,
// INPUT= ,
// VOLI= ,
// OUTPUT= ,
// VOLO=
//UPDATE1 EXEC PGM=WEEKLY,REGION=&REGSIZE
// INPUT DD DSNAME=&INPUT,UNIT=TAPE,VOLUME=SER=&VOLI,
// DISP=OLD
//OUTPUT DD DSNAME=&OUTPUT,UNIT=TAPE,VOLUME=SER=&VOLO,
// DISP=(,KEEP)
```

More detailed information on symbolic parameters can be found in the manual *OS/390 MVS JCL Reference*, GC28-1757.

2.11.3 Using Cataloged Procedures

To use a cataloged procedure, the procedure name is coded on the EXEC statement, and any necessary symbolic parameters are specified. To use the above example of a cataloged procedure, code:

```
//STEP1 EXEC UPDATE,REGSIZE=2M,INPUT=dsname1,  
//          VOLI=serial1,OUTPUT=dsname2,VOL0=serial2
```

It is frequently necessary to modify one or more parameters in a cataloged procedure in order to make use of it. A procedure can be modified by assigning values to or nullifying symbolic parameters contained in the procedure; by overriding, adding, or nullifying parameters on EXEC statements in the procedure; or by adding or overriding DD statements in the procedure.

2.11.4 Assigning Values to and Nullifying Symbolic Parameters

Many procedures contain symbolic parameters which must be assigned values or nullified unless the default values are satisfactory. For example, to use the NIH Computer Center procedure DSLIST to list the third data set, userid.DATA on a 3490 tape:

```
// EXEC DSLIST,NAME='userid.DATA',STORAGE=TAPE,SEQ=3
```

Symbolic parameters need not be coded in any specific order when they are assigned values or nullified.

2.11.5 Modifying Parameters on the EXEC Statement

A cataloged procedure must contain one or more EXEC statements, one for each program to be executed. Each EXEC statement can contain keyword parameters to allow conditional execution of the step, to specify a region size for the step, or to pass information to the program executed in the step. These EXEC statement parameters are COND, REGION, and PARM. For an explanation of the meaning and use of these parameters, see section 2.5.4.

To add one of these parameters to a step in a procedure, or to override a parameter that already exists, the user must code:

```
parameter.procstepname=value
```

on the EXEC statement invoking the procedure. The *parameter* is one of the keyword parameters listed above, and *procstepname* is the name of the step in the procedure to which the parameter will apply. For a multi-step procedure, all overrides or additions to one step must be coded before the overrides or additions to a succeeding step.

The following EXEC statements would be valid for a three-step procedure named IIIPROC with stepnames GO1, SORT2, and PRINT3:

```

//S1 EXEC IIIPROC,REGION.GO1=2M,COND.SORT2=(16,EQ),
//  PARM.PRINT3=NUMBER

//S1 EXEC IIIPROC,REGION.GO1=1200K,REGION.SORT2=300K,
//  PARM.SORT2='SIZE=204800',COND.PRINT3=(0,NE,SORT2)

```

The following EXEC statement is invalid because the parameters are not in procedure step order:

```

//S1 EXEC IIIPROC,REGION.SORT2=400K,REGION.GO1=660K,
//  COND.PRINT3=(4,EQ),COND.GO1=(4,NE)

```

If an EXEC statement parameter is coded without a qualifying stepname, it will apply to the entire procedure. The parameter PARM, when used without a stepname, will apply to the first step and nullify all succeeding PARM fields. COND applies to every step in the procedure and overrides any COND information given in the procedure. REGION applies to every step and overrides all REGION values coded in the procedure.

2.11.6 Overriding or Adding DD Statements

It may be necessary to override DD statements in a procedure or add DD statements to a procedure. These DD statements may:

- define data card images in the input stream
- define a data set for an ABEND dump
- describe a data set on tape or disk
- define a printer
- override any DD statements or parameters on DD statements already in the procedure.

An overriding or added DD statement must use the format:

```
//procstepname.ddname DD parameters
```

where *procstepname* identifies the procedure step that the user is overriding and *ddname* is the DD statement in that procedure step. The following DD statement is an example:

```
//GO.MASTIN DD UNIT=TAPE
```

MASTIN is the name of the DD statement and *GO* refers to the name of the procedure step to which this override or addition applies.

Place overriding DD statements in the following order after the EXEC statement that calls the procedure:

- Overriding or added statements that explicitly specify the step being overridden can appear in any order.
- Overriding and added statements that do not explicitly specify the step are applied to the step named in the previous overriding or added DD statement. If no previous override statement named a step, then they are applied to the first step in the procedure.
- Place the overriding statements for each procedure step in the same order in which the procedure steps are specified.

If the user adds a ddname, it must be different from any ddname already in the procedure step.

The overriding DD statement should contain only those keyword parameters you want to replace, add or nullify. Code the entire parameter with its new value on the overriding DD statement in any order. To nullify parameters, code the keyword parameter followed by an equal sign.

Normally, parameter values in the override DD statement replace the parameter values in the procedure. The DCB parameter is the only exception to this rule. You can add or replace subparameters in the DCB without causing the entire field to be replaced. DCB=BLKSIZE=11438 on an override statement will merge this block size with the other DCB characteristics already coded for the DD statement in the procedure.

In the DD statement below, both the UNIT and SPACE parameters are nullified on the OUTPUT DD statement in the GO step.

```
//GO.OUTPUT DD UNIT=,SPACE=
```

The DCB parameter is nullified only when each subparameter is nullified.

```
//GO.OUTFILE DD DCB=(RECFM=,LRECL=,BLKSIZE=)
```

Certain parameters are mutually exclusive and coding one of them on an override statement will cause the other to be nullified. Code only one parameter in a mutually exclusive pair. DISP and SYSOUT is an example of a mutually exclusive pair, as is DSNAME and DUMMY.

If you use a procedure containing a SPACE parameter that specifies RLSE or a value for the secondary parameter and if the DD statement refers to an existing disk data set, you should nullify the SPACE parameter (SPACE=). If the SPACE parameter is

not nullified, the value for the secondary parameter will be applied to the data set, and any unused space in the existing data set will be released.

Override DD statements that have * or DATA coded in the operand field completely negate any parameters coded in the procedure's DD statement.

To be sure of the technical correctness of overrides to procedures maintained by the NIH Computer Center, use the JCLSCAN procedure (see section 2.13.6) before running the job. This will provide a listing of the procedure and the accompanying override statements without actually running the job. The NIH Computer Center reserves the right to change, without warning, any of its procedures at any time if the change will not affect published usage of the procedure.

2.11.7 Instream Procedures

An instream procedure is a set of job control language statements in the form of card images that have been placed in the input stream of a job. Instream procedures can be extremely useful in testing procedures that are to be placed in a procedure library and in setting up a set of JCL for repeated use during a single job. An instream procedure can be executed any number of times during the job in which it appears, and up to fifteen uniquely named instream procedures can appear in one job. An instream procedure consists of one or more steps called procedure steps. Each procedure step consists of an EXEC statement and DD statements.

The instream procedure can follow the JOB statement or the JOBLIB DD statement. It must precede the EXEC statement that invokes it.

An instream procedure is subject to the same restrictions as a regular procedure. It must not contain JOB statements, delimiter statements, null statements, JOBLIB DD statements, or DD statements with * or DATA coded in the operand field.

To use an instream procedure, the procedure name is coded on the PROC statement, and any necessary symbolic parameters are specified. The user may override parameters, modify DD statements contained in the procedure, and add DD statements to the procedure.

An instream procedure must be preceded by a PROC statement and followed by a PEND statement.

```
//BACKUP PROC DSNAME= ,
// STORAGE=DISK2YR ,
// TAPE= ,
// UNIT=TAPE
//STEP1 EXEC PGM=COPYIT,REGION=500K
//DD1 DD DSNAME=&DSNAME,DISP=SHR,UNIT=&STORAGE
//OUT DD DSNAME=&DSNAME,DISP=(NEW,KEEP) ,
// UNIT=&UNIT,VOLUME=SER=&TAPE,DCB=DEN=&DEN
//STEP2 EXEC PGM=SCRATCH,COND=(0,NE)
```

```

//DEL      DD  DSNAME=&DSNAME , DISP=( OLD , DELETE ) ,
//          UNIT=&STORAGE
//          PEND

```

More detailed information on symbolic parameters can be found in the manual *OS/390 MVS JCL Reference*, GC28-1757.

The operands field of the PROC statement either contains values for symbolic parameters or is blank. The name and comments fields on the PEND statement are optional.

The above procedure may be placed within the job stream and may be invoked any number of times. The following example illustrates how the user might override and add to the DD statements of the instream procedure listed above:

Examples

```

//BEGIN      EXEC  BACKUP , DSNAME='userid.OLDMAST' ,
//          TAPE=123456
//NEXT      EXEC  BACKUP , DSNAME='userid.OLDDATA' ,
//          STORAGE=TEMP , TAPE=123456
//STEP1.OUT  DD   LABEL=( 2 , SL )
//STEP2.DEL  DD   DISP=( OLD , KEEP )

```

2.11.8 INCLUDE Groups

An INCLUDE group is a set of JCL statements that may be included in a JCL job stream. The JCL statements must be saved as a member of a cataloged procedure library that is specified on a JCLLIB statement. For more information on JCLLIB, see section 2.11.1. An INCLUDE group may not contain any of the following:

- JOB statements
- PROC and PEND statements
- JCLLIB statements
- JES2 control statements (statements beginning with a /*)
- DD * and DD DATA statements.

For example, if the following statements are saved as member PRINT in a PDS named userid.MYLIB:

```

/* THIS STEP PRINTS A DATA SET .
//PRINT EXEC COPY
//SYSUT1 DD DSNAME=&OLDDATA , DISP=( OLD , DELETE )
//SYSUT2 DD SYSOUT=A

```

Then the user submits the following job stream:

```
//III JOB ( ), USERNAME
// *
//LIB JCLLIB ORDER=userid.MYLIB
// *
//STEP1 EXEC PGM=MAIN
:
:
//OUTPUT DD DSNAME=&OLDDATA, DISP=(NEW, PASS)
// *
// INCLUDE MEMBER=PRINT
```

Any output written by program MAIN to the temporary data set name &OLDDATA is passed to the next step (which was INCLUDED) and printed on the central printers.

2.12 Printing

This section tells how to produce printed output on Titan.

2.12.1 JCL for Printing

You can control the appearance of output by using the OUTPUT JCL statement, the DD statement, or the /*JOBPARM statement. The OUTPUT JCL statement gives you the most parameters for specifying characteristics of the output.

2.12.2 Output Delivery

All output that prints on the Titan central printers must have a box number associated with the job. The box number can be associated with your userid or specified in the JCL.

When you are initially registered on Titan, the box number field is set to NOBX or NONE unless your account sponsor has provided a box number. To see if a box number is associated with your userid, go to the Titan Customer Locator web page at <http://titan.nih.gov/locator>. You can change the box number associated with your userid from this web page.

To specify a box number in your JCL, use the ROOM parameter of the OUTPUT JCL or /*JOBPARM statement or use the /*BOX statement.

If you try to run a job on Titan with a box number of NOBX, the job will fail with a JCL error. If you do not plan to use the central printers, you should set the box number to NONE.

If the job is run offsite and printed on a VPS or other non-central printer, the NOBX designation will not cause a problem. Any output that prints on a central printer but has a box number of NOBX, NONE, zero, or blank will be disposed of and the user will be charged for printing.

Output designated for Parklawn boxes ("P" preceding the box number) will be delivered to the CIT Offsite Distribution Center in the Parklawn building (room 2B-70). See the *Titan User's Guide* for messenger service information.

2.12.3 Forms Available

The following printing forms are available:

- standard form
- cut-sheet forms
- labels
- user-supplied forms

2.12.4 Standard Form

The standard form offers the fastest turnaround. It is the form used if you do not specify a form in your JCL. The following chart shows the defaults for the standard form.

FORM	Width	Length	Character Set	Lines per inch	Printer	Description
STD	11 inches	8 1/2 inches	GT12 (12-pitch)	8	IBM 3900 laser	continuous form paper, fan-fold, single-sided

Paper size is given with easy-strip margins removed. Pitch refers to the number of characters per inch.

You can change the number of lines per inch by using the PAGEDEF parameter on the OUTPUT JCL statement. The default of 8 lines per inch is equivalent to a PAGEDEF of 808. The last 2 digits specify the number of lines of text per inch and the first digit specifies the length of the page in inches. Use 806 for 6 lines per inch and 810 for 10 lines per inch.

The maximum lines per page for each LPI is shown below.

Maximum Lines per Page

Lines per Inch	STD form
6	45
8	60
10	75

You can change the character set by using the CHARS parameter on the OUTPUT JCL or DD statement. For a list of available character sets, see

<http://datacenter.cit.nih.gov/print-ug>

The recommended character pitch for each LPI is as follows:

Recommended Character Pitch

Lines per Inch	Recommended Character Pitch
6	10, 12, or 15 pitch
8	12 or 15 pitch
10	15-pitch

The chart below shows the maximum characters per line for each pitch. Additional characters are truncated.

Maximum Characters per Line

Form	10 pitch	12 pitch	15 pitch
STD	110	132	165

The laser printers cannot be used to do printing on pressure sensitive labels; overprinting with any character except the underline; or printing within 1/2 inch of the top or bottom of a page.

2.12.5 Cut-Sheet Forms

Cut-sheet forms offer double-sided printing, 3-hole punch, and other options. You must specify the FORMS and FORMDEF parameters in your JCL. The following chart shows the defaults for cut-sheet forms.

FORMS	FORMDEF	Description	Width	Length	Char Set	Lines per inch	Printer
900	LS00	landscape, single-sided	11 inches	8 1/2 inches	GT12	8	IBM 3160 laser
	LD00	landscape, double-sided					
999	LS99	landscape, 3-hole punch, single-sided	11 inches	8 1/2 inches	GT12	8	IBM 3160 laser
	LD99	landscape, 3-hole punch, double-sided					
900	PS00	portrait, single-sided	8 1/2 inches	11 inches	GT12	8	IBM 3160 laser
	PD00	portrait, double-sided					
	LH1	NIH letter head					
999	PS99	portrait, 3-hole punch, single-sided	8 1/2 inches	11 inches	GT12	8	IBM 3160 laser
	PD99	portrait, 3-hole punch, double-sided					

FORMDEF LH1 uses the ISO/ANSI carriage control "2" to skip past the heading and begin at 1 and 3/4 inches below the top.

You can change the number of lines per inch by using the PAGEDEF parameter on the OUTPUT JCL statement. Use the charts below to determine the correct PAGEDEF parameter.

Landscape printing

Lines per Inch	PAGEDEF
6	806 (the default)
8	808
10	810

For landscape printing, the last 2 digits specify the number of lines per inch and the first digit specifies the length of the page.

Portrait printing

Lines per Inch	PAGEDEF
6	1106 (the default)
8	1108
10	1110

For portrait printing, the last 2 digits specify the number of lines per inch and the first 2 digits specify the length of the page.

The maximum lines per page for each LPI is shown below.

Maximum Lines per Page

Lines per Inch	Landscape	Portrait
6	45	60
8	60	80
10	75	100

When printing using FORMDEF=LD99 and PAGEDEF=808, use LINECT=59 to avoid printing over the holes on the back of each page.

You can change the character set by using the CHARS parameter on the OUTPUT JCL or DD statement. For a list of available character sets, see

<http://datacenter.cit.nih.gov/print-ug>

The recommended character pitch for each LPI is as follows:

Recommended Character Pitch

Lines per Inch	Recommended Character Pitch
6	10, 12, or 15 pitch
8	12 or 15 pitch
10	15-pitch

The chart below shows the maximum characters per line for each pitch. Additional characters exceeding the maximum are not printed.

Maximum Characters per Line

Form	10 pitch	12 pitch	15 pitch
900, 999 landscape	108	129	161
900 portrait	83	100	124
999 portrait	78	94	117
900 LH1	84	101	126

2.12.6 Labels

Labels are pre-gummed, in one column, and attached to sheets of waxed paper. Samples are available at Output Distribution Services. They are printed on the impact printers.

You must specify the FORMS and FCB parameters in your JCL. Column 1 is reserved for carriage control characters. The data must have a valid carriage control character on the first line of each label. The following chart shows the defaults for labels. A record format (RECFM) ending in A indicates that the data set has ISO/ANSI carriage control (e.g., RECFM=FA).

Labels for Impact Printing

FORMS	Width	Depth	Char s Per Line	Beginning Column for Text	Lines per Label		Carriage Control – Start at Line 2	Carriage Control – Start at Line 1
					6 lpi	8 lpi		
L2S1	2 3/4	7/16	26	2	2	3		1, 2, 3
L3S1	3 1/2	7/16	34	2	2	3		1, 2, 3
L3M1	3 1/2	15/16	34	2	5	7	1, 2	4, 5
L4L1	4	1 7/16	38	2	8	11	1	4
L5L1	5	1 7/16	49	2	8	11	1	4

Small labels are 7/16 inch deep. Medium labels are 15/16 inch deep. Large labels are 1 7/16 inches deep. Small labels always start printing on line 1. Medium and large labels can start printing on line 1 or line 2, depending on which carriage controls you use.

Small labels have 3 labels per logical page, so use carriage control 1 for the first label, 2 for the second label, 3 for the third label, 1 for the fourth label, and so on.

Medium labels have 2 labels per logical page, so to start on line 2, use carriage control 1 for the first label, 2 for the second label, 1 for the third label, 2 for the fourth label, and so on. To start on line 1, alternate the number 4 and 5 instead of 1 and 2.

Large labels have 1 label per logical page, so to start on line 2, use carriage control 1 for every label. To start on line 1, use carriage control 4 for every label.

Place the carriage control in column 1. The text of the label should begin in column 2.

If possible, try to design your labels to use one less line than the maximum allowed for your label's LPI to minimize the risk of losing the last line.

Form Control Buffers (FCBs) to Use With Labels

FORM	FCB for 6 lpi	FCB for 8 lpi
L2S1	LS06	LS08
L3S1	LS06	LS08
L3M1	LM06	LM08
L4L1	LL06	LL08
L5L1	LL06	LL08

The naming convention for these FCB's is that they are of the form *ldnn* where:

- l stands for labels.
- d is the depth of the label:
 - S for 7/16 inch (small)
 - M for 15/16 inch (medium)
 - L for 1 7/16 inches (large)
- nn is the number of lines printed per inch.

Using the WYLABEL procedure

The WYLABEL procedure prints labels from a single column of data. For information on WYLABEL, see section 4.7.3.

2.12.7 User-Supplied Forms

The NIH Computer Center provides for printing on custom-designed user forms. Such forms include pre-printed invoices, multi-ply forms, and card stock. These forms use the impact printers and the jobs may take up to 24 hours to be printed. Only the SN10 character set is available on the impact printers.

A job must be printed on an impact printer if it requires:

- printing that extends across page boundaries
- overprinting with a character other than an underline
- printing on pressure sensitive labels or multi-part forms

Registering and Supplying Forms

Each user-supplied form must be registered with Output Distribution Services before it is used for the first time. Jobs that request a non-registered form will not print or will be printed incorrectly. A specific form needs to be registered only once. If the

form changes significantly (e.g., size, pre-printed material, number of parts), it should be re-registered. Once registered, a supply of the form must be delivered to Output Distribution Services each time it is used. Only a small supply can be maintained because of space limitations and fire regulations.

Output Distribution Services requires the following information when a form is registered:

Specification	Description
width	in inches (includes perforated margins and backing for labels)
usable width	how much of the width may be used for printing
length	in inches for a page or sheet of labels
parts	number of plies
FCB	the name of the Forms Control Buffer (FCB) to be used for the form. Several FCBs are provided by the NIH Computer Center and one of these, or a user-supplied FCB, may be specified. When requesting a form, the user must include an FCB in the job control language.
character set	the name of the character set to be used with the form. See section 2.12.8 for information on character sets.
alignment data set	the name of a data set that can be used to verify form alignment. This data set should contain valid carriage control characters.
sample page	a sample listing of the alignment data set printed with proper alignment. To make this easier, the form may be pre-registered by contacting Output Distribution Services.
person responsible	name, address, and telephone number of the individual responsible for the form

At the conclusion of the registration process, the person who registered the form will be given a four-character form name. This name must be specified in the FORMS= parameter of the user's JCL when the form is to be used.

2.12.8 Character Sets

The impact printer uses only the SN10 character set. The laser printers use SN10 and the other character sets shown below. The last two characters in the name indicate the

pitch (number of characters per inch). Tri-pitch fonts (TR) use three character widths to accommodate characters of different widths (*w* and *i*, for example.)

The character sets available for the laser printers are:

Tri-pitch	BITR, BRTR, DOTR
Courier	CE10, CE12, CI15, CO10, CR10
	CN10
13.3 pitch	D225, D226, D227
Essay	EBTR, EITR, ESTR
Gothic	GB10, GB12, GI12, GR10, GT10, GT12, GT15
Letter Gothic	LB12, LR12
OCR	OA10
Orator	OB10, OR10
Prestige	PB12, PI12, PR10, PR12
Roman	RT10
Standard	SE10, SE12, SN10, SN12, SN15
Serif	SB12, SI10, SI12, SO12, ST10, ST12, ST15
Script	SR12
Symbol	SY10, SY12
	TN10

The character sets D225, D226, D227, and CI15 are available on the cut-sheet laser printers only:

The OCR character set (OA10) is designed to be scanned by an optical character reader.

Up to 4 character sets may be used in a document printed on the cut-sheet laser printer.

Up to 2 character sets may be used in a document printed on the continuous form laser printer. To view the character sets go to:

<http://datacenter.cit.nih.gov/print-ug>

Hexadecimal Tables for Character Sets

To view tables showing the hexadecimal character representations for all available characters in each character set and keyboard diagrams, go to the Web address above and click "Hexadecimal Tables."

2.12.9 Using Carriage Control

Carriage control characters appear in column 1 of the data set and tell the printer how much to advance the printer carriage. They are used to achieve special effects, such as

change of character sets within a line of printing and underscoring characters. There are two kinds of carriage control: machine carriage control and ISO/ANSI carriage control:

- Machine carriage control uses unprintable hexadecimal characters in column 1 of each record of the data set to control vertical spacing. It is the carriage control that printers actually use to control vertical spacing when printing on a form.
- ISO/ANSI carriage control uses printable characters in column 1 of each record of the data set to control vertical spacing. The ISO/ANSI carriage control characters are translated into machine carriage control by the JES2 component of the operating system before the output is printed.

A record format (RECFM) ending in A indicates that the data set has ISO/ANSI carriage control; one ending in M indicates machine carriage control. If a data set does not contain carriage control, the RECFM should not include A or M (e.g., RECFM=FB), and column one of each record of the data set will be treated as text.

When carriage control characters are used for controlling output spacing, two types of positioning can be used: line-oriented and channel-oriented. Line-oriented positioning specifies the number of lines to move the carriage before or after the line is printed.

The line-oriented carriage control characters are defined by the system, and are NOT specified within the Forms Control Buffer (FCB), which defines vertical spacing for a page of printed output.

In channel-oriented positioning, the FCB relates channel codes (1-9, A-C) to specific lines on a form. Channel-oriented carriage control characters initiate a vertical "skip" down the lines of a page. To terminate the skip, a channel code corresponding to the carriage control character must be coded in the FCB. This channel code marks the line on the form where the skip will terminate. For example, if the FCB defines channel 8 to relate to line 55 on a form, and a carriage control character of 8 is encountered in the output, the form will automatically advance to line 55 before printing resumes. As another example, if the FCB defines channel 11 ("B") to be associated with lines 14 and 45 on a form, and a carriage control character of B is encountered in the output, the form will automatically advance to either line 14 or 45, whichever is encountered first.

The JES2 component of the operating system requires that Channel 1 be defined in all FCBs. JES2 "skips to Channel 1" prior to printing the header and trailer pages on the output. For output other than labels, Channel 1 is always print line 4 for 6 lines-per-inch density, line 5 for 8 lpi, and line 6 for 10 lpi. Other channels may be set at vertical positions above Channel 1 for impact printing, but may not be set above Channel 1 in FCBs intended for the laser printers. For labels, Channel 1 should be defined near the top of the label and Channel 10 should be defined as the last print line before the perforation on a page of labels.

Line-Oriented Carriage Control Characters

ISO/ANSI CODE (EBCDIC)	ACTION BEFORE WRITING PRINT LINE
-----------------------------------	---

blank	Advance one line
0	Advance 2 lines
-	Advance 3 lines
+	Do not advance

MACHINE CODE (HEX)	PRINTER ACTION
-------------------------------	-----------------------

01	Print, then do not advance
09	Print, then advance 1 line
11	Print, then advance 2 lines
19	Print, then advance 3 lines
0B	Advance 1 line, do not print
13	Advance 2 lines, do not print
1B	Advance 3 lines, do not print
03	No operation

Channel Oriented Carriage Control Characters

ISO/ANSI CODE (EBCDIC)	ACTION BEFORE WRITING PRINT LINE
-----------------------------------	---

1	Skip to Channel 1
2	Skip to Channel 2
3	Skip to Channel 3
4	Skip to Channel 4
5	Skip to Channel 5
6	Skip to Channel 6
7	Skip to Channel 7
8	Skip to Channel 8
9	Skip to Channel 9
A	Skip to Channel 10
B	Skip to Channel 11
C	Skip to Channel 12

MACHINE CODE (HEX)	PRINTER ACTION
---------------------------	-----------------------

89	Print, then skip to Channel 1
91	Print, then skip to Channel 2

MACHINE CODE (HEX)	PRINTER ACTION
99	Print, then skip to Channel 3
A1	Print, then skip to Channel 4
A9	Print, then skip to Channel 5
B1	Print, then skip to Channel 6
B9	Print, then skip to Channel 7
C1	Print, then skip to Channel 8
C9	Print, then skip to Channel 9
D1	Print, then skip to Channel 10
D9	Print, then skip to Channel 11
E1	Print, then skip to Channel 12
8B	Do not print, skip to Channel 1
93	Do not print, skip to Channel 2
9B	Do not print, skip to Channel 3
A3	Do not print, skip to Channel 4
AB	Do not print, skip to Channel 5
B3	Do not print, skip to Channel 6
BB	Do not print, skip to Channel 7
C3	Do not print, skip to Channel 8
CB	Do not print, skip to Channel 9
D3	Do not print, skip to Channel 10
DB	Do not print, skip to Channel 11
E3	Do not print, skip to Channel 12

Computer Center Forms Control Buffers (FCBs)

The following FCBs are provided by the NIH Computer Center:

FCB Name	Form Length (in inches)	Lines per inch
806	8 1/2	6
808	8 1/2	8
810	8 1/2	10
1106	11	6
1108	11	8
1110	11	10

These FCBs may be used with any registered form that is of the correct length. Thus, for an 11-inch form, 1106, 1108, or 1110 is valid. Each of these FCBs has the following carriage control characters defined:

Carriage Control Character	Action
1	Skip to the top of the next page.
2	Skip to the next half (middle or bottom) of the

**Carriage Control
Character**

Action

3	page. Skip to the next third (one-third or two-thirds) of the page.
9	Skip to the next-to-the-last-line of the page.

Registering User Forms Control Buffers

Users can register forms control buffers (FCBs) that differ from those provided by the NIH Computer Center. Contact Output Distribution Services to perform the FCB registration. To complete this registration, provide the following information:

form length	the length of the form, specified in inches
vertical density	the number of lines per inch that will be used for the FCB. The values available are 6, 8, and 10 lines per inch. Please note that a separate FCB must be registered for each density that will be used with a given form.
channel codes	the channels to be defined and the line numbers that are to be associated with them. Channel 1 must be defined as print line 5.
person responsible	name, address, and telephone number of the individual responsible for the FCB

At the conclusion of the registration process, the person who is responsible for the FCB will receive a four-character FCB name. This FCB name must be specified in the user's JCL when the FCB is to be used.

2.12.10 Table Reference Characters (TRC)

When multiple character sets are used in a single document, table reference characters (TRCs) and the CHARS parameter indicate which portions of the output are to be printed in each character set. The DCB subparameter OPTCD=J must be used to describe the data set. The TRCs occupy a column to the left of the text, following the optional carriage control column. A TRC of blank or "0" selects the first font; "1" selects the second font.

The following example uses "0" and "1" in the TRC column to print a small table in GT15 when the rest of the text is printed in PR12.

```
0Often within output, it is desirable to be able to switch to a
0smaller character set such as the Gothic Text 15-Pitch which allows
0a table that might be too wide to fit on the page.
0
1Data Group 1      Data Group 2      Data Group 3      Data Group 4
lexercised        exercised        sedentary         sedentary
```

lnormal weight	overweight	normal weight	normal weight
lnon-smoker	smoker	non-smoker	smoker
lnormal bp	normal bp	normal bp	hypertensive

The printed output would look similar to the following:

Often within output, it is desirable to be able to switch to a smaller character set such as the Gothic Text 15-Pitch which allows a table that might be too wide to fit on the page.

Data Group 1	Data Group 2	Data Group 3	Data Group 4
exercised	exercised	sedentary	sedentary
normal weight	overweight	normal weight	normal weight
non-smoker	smoker	non-smoker	smoker
normal bp	normal bp	normal bp	hypertensive

The DD statement for the output data set might be:

```
//GO.OUTPUT DD SYSOUT=A,
// DCB=(BLKSIZE=130,RECFM=F,OPTCD=J),CHARS=(PR12,GT15)
```

To mix two character sets within a line, TRCs and carriage control must be used. As the next example shows, a "+" carriage control character in column 1 can cause a line printed in one character set to be overlaid with characters from another.

```
0This is a test of trc characters and          fonts on the
+1                                dual
0continuous form printer.  It will show how trc characters
1can be used to create font changes within lines as well
0as between lines.
```

The printed output would look similar to the following:

This is a test of trc characters and **dual** fonts on the continuous form printer. It will show how trc characters **can be used to create font changes within lines as well as between lines.**

The following DD statements describe the output data set:

```
//SYSPRINT DD SYSOUT=A,
// DCB=(BLKSIZE=80,RECFM=FA,
// OPTCD=J),CHARS=(SN10,GB10)
```

The record format (FA) indicates that the data set contains carriage control characters in column 1. Remember that the program must write the TRCs in front of each line of the output.

2.12.11 Hardcopy Utility on Titan ISPF

The hardcopy utility is a series of panels that can be accessed by entering Option L.7 or 3.6 from the CIT/Titan Primary Option Menu. With this utility you can access JCL for printing on the printers described in this section. JCL for multiple-up, tilt (rotate), and duplex has been included to make printing your data sets as easy as possible. To use the hardcopy utility:

- Go to the CIT/Titan Primary Option Menu.
- Enter L.7 (a CIT-written panel that offers an easy way to get additional features such as 4-up (4 page images on a physical page), duplex, etc.)

or

enter 3.6 (the standard IBM panel)

- Provide the required information.
- Press Enter.

If you use panel L.7 and wish to change any of the hard-coded JCL, change SUBMITTING=Y to N. You will receive the panel with the JCL that can be altered. At this point, you can make the desired changes. Then, type SUB at the command line to submit the job. The JCL changes will revert to the hard-coded JCL the next time you use it.

2.12.12 Network Printing Services

Users can print output from Titan to a local networked printer using the VPS printing service. For more information on using VPS, see:

<http://silk.nih.gov/silk/vps>

Macintosh Printing Using PrintShare

PrintShare, a NIHnet service developed and supported by CIT, allows networked PC, Unix, and OS/390 System users to print to the nearest networked AppleTalk printer.

PrintShare can be used with VPS to send OS/390 mainframe print jobs, such as Administrative Data Base (ADB) print reviews and WYLBUR listings, directly to a networked AppleTalk printer. For further information, or to register an AppleTalk printer for PrintShare, call the NIH Help Desk or visit:

<http://silk.nih.gov/silk/vps>

2.12.13RJE Workstation Facilities

Remote Job Entry Workstations are located at users' sites and communicate with the NIH Computer Center mainframes over telephone lines. Job output as well as individual SYSOUT data sets can be directed to a remote workstation. Because the remote workstations are obtained by individual organizations, they vary in their facilities, services, and regulations.

For more information on remote workstations, see the *Titan Users Guide*. To direct your output to a remote workstation, see the DEST parameter of the DD statement (section 2.6.4) or of the OUTPUT JCL statement (section 2.7.4).

2.12.14VTAM Printers

VTAM printers are part of the SNA network and are located at users' sites. They are connected to controllers that communicate with the central facility via dedicated lines. They can be defined as RJE printers and output routed to them use their VTAM node name. See the *Titan User's Guide* for information about VTAM printers.

2.13 Submitting Jobs

The following section describes some methods of submitting jobs to Titan. An additional method, the DSSUBMIT procedure, is described in section 8.2.1.

2.13.1 Interactive Job Submission

Titan supports several interactive teleprocessing systems (e.g., TSO, WYLBUR, ADABAS, and Model 204), which include capabilities for job submission via remote terminals. For details on submitting batch jobs, refer to the specific documentation for the teleprocessing system to be used. For submitting jobs through the Web, see section 2.13.2.

2.13.2 Batch Job Submission from Another Batch Job

A batch job can submit another batch job. See section 8.2.1.

2.13.3 Batch Job Submission through the Web

Users can submit their batch jobs through the World Wide Web using the Titan Submit facility of SILK (Secure Internet LinKed) technologies.

Submitting a Single Job Using Web Submit

Open your Web browser software to:

<http://titan.nih.gov>

- Select **Web Submit**.
- Follow the on-screen prompts.

-
- Click the **Submit** button.

After you click on the submit button, the data set will be submitted and the job number will be displayed.

If the job was rejected, information will also be displayed. To find why the job was rejected, look at the printed output or fetch the job through WYLBUR or TSO.

Submitting a Single Job Using TSO Commands

Open your Web browser software to:

<http://titan.nih.gov>

- Select TSO Commands.
- Select the **submit** command.
- Enter a data set name.
- Click the **Enter** button.

You will be prompted for your Titan userid and password before the job is submitted.

Information Supplied by the System

- If you do not code USER= on the JOB statement, the job will be run under the userid supplied in the Web prompt.

Information Supplied by the User

- If the userid coded on the USER= parameter of the JOB statement is the same as the userid supplied in the Web prompt, the system will provide the proper password.

The data set being submitted via SILK Web's submit button **must be RECFM=FB with LRECL=80**. The data set **cannot** be WYLBUR edit format. The userid specified in the browser security window must have authority at least to read the data set. After you click on the submit button, the data set will be submitted and the job number will be displayed.

If the job was rejected, information will also be displayed. To find why the job was rejected, look at the printed output or fetch the job through WYLBUR or TSO.

Scheduling Recurring Work

Titan Job Scheduler allows users to schedule recurring work to run automatically. This Web-based facility gives you control over setting up, displaying, and modifying your batch job entries. You can schedule a job to be submitted once or on a continuing basis. Simply provide the name of your JCL data set and the schedule for running it.

After that, your job is run automatically on the date and time you requested with no further action on your part.

To use the Job Scheduler, go to:

<http://titan.nih.gov>

- Select **Job Scheduler**.
- Follow the on-screen prompts.

For further information on Titan SILK Web technologies, refer to the *Titan User's Guide*, or visit:

<http://titan.nih.gov>

2.13.4 Batch Job Submission through FTP

You can use the FTP command SITE FILETYPE=JES to transmit your job to Titan. For information on this method, see *Network Access to the Titan System*, which you can order through CIT Publications (see section 1.3).

2.13.5 Batch Job Cancellation

There may be instances when a user wishes to cancel a batch job after it has been submitted for processing. To cancel a batch job, use one of the following procedures:

- Interactive - Batch jobs may be cancelled from Interactive Output Facility (IOF) under TSO.
- Web - Go to:

<http://titan.nih.gov>

Select TSO Commands and select **cancel** from the Command list box.

- High-Speed Remote Job Entry (RJE) - RJE operators can cancel batch jobs submitted from, or routed to, their work station by entering a properly coded JES2 Control Statement.

2.13.6 Job Syntax Checking

JCL is checked in two steps.

-
- Conversion processing is done when the job first enters the job queue. During conversion, syntax errors are checked, and all JCL (including expanding procedures) is resolved.
 - Later, at interpretation, the system detects other errors, including misplaced JCL statements.

If a job fails during conversion because of JCL errors and the job is corrected and resubmitted, logical errors might still be found during interpretation. Users will be informed of failures if the jobs in question have the NOTIFY attribute. This is particularly useful for jobs which are run overnight. The syntax errors will be detected, and the job will print immediately or be placed in the OUTPUT HOLD queue giving the user a chance to correct the JCL and resubmit the job. If a job is directed to OUTPUT HOLD, the output from the conversion phase will be available when that phase is completed. When the converter encounters an error, a user signed onto WYLBUR or TSO will be notified if the job has the NOTIFY attribute.

To check JCL without running a job, you can use TYPRUN=SCAN or use JCLSCAN:

- The TYPRUN=SCAN parameter on the JOB statement, described in section 2.2.4, checks the JCL in conversion.
- The JCLSCAN procedure, described in section 8.1, checks the JCL in conversion and interpretation.

2.13.7 Executing a Job at Another Node

You can submit a job on Titan to execute at another node. The following restrictions apply:

- The other node must be defined to NIH as a JES2 NJE node. That is, it must be an OS/390 JES2 or JES3 node, a VM RSCS node, or a VSE POWER node.
- You must have an authorized userid at both the sending and receiving nodes.
- Any job that is submitted to Titan must use the USER and PASSWORD parameters on the JOB statement to specify the Titan userid and password.
- To view the job in OUTPUT HOLD on Titan rather than printing immediately, add a `/*ROUTE PRINT NIHJES2 .HOLD` statement.
- If you run a job at another NJE node and then route the job to print on Titan, you must add a `/*JOBPARM ROOM=bbbb` statement as described in section 2.3.1.

- The job must contain a /*NETACCT statement as described in section 2.3.1. The output will be rejected by Titan unless the userid that appears on the /*NETACCT statement has been authorized to the facility class.

For a table of nodes defined to Titan, see section 7.2.1

The following example shows a job submitted on Titan to be run at a node named “nodename.” This job uses a /*XMIT statement. The /*NETACCT statement must appear before the /*XMIT statement.

Example

```
//jobname JOB (Titan JOB statement)
/*NETACCT titanuserid
/*XMIT nodename
//njob JOB (JOB statement for receiving node)
/*JOBPARM ROOM=bbbb
/*ROUTE PRINT NIHJES2.HOLD
(any other control statements required by receiving host)
//stepname EXEC ...
```

The next example uses a /*ROUTE XEQ statement instead of the /*XMIT statement:

Example

```
//njob JOB (JOB statement for receiving node)
/*ROUTE XEQ nodename
/*NETACCT titanuserid
/*JOBPARM ROOM=bbbb
/*ROUTE PRINT NIHJES2.HOLD
(any other control statements required by receiving host)
//stepname EXEC ...
```

The next example assumes that you submit a job while you are logged on at another node and that the job runs at that node. If you want to have this job print on the Titan central printers, you should add the following statements:

Example

```
//njob JOB (JOB statement for other node)
/*NETACCT titanuserid
/*JOBPARM ROOM=bbbb
/*ROUTE PRINT NIHJES2
(any other control statements required by other node)
//stepname EXEC ...
```

2.13.8 Remote Job Submission

Titan supports high-speed remote job entry (RJE) workstations located at user sites that communicate with the central facility via telephone lines. You can use the following statements and parameters.

USER

The userid and password are required in the JOB statement for all jobs submitted from an RJE node. The format for the parameters is:

```
,USER=userid,PASSWORD=password
```

SIGNON

The SIGNON control statement must be the first statement entered at the RJE workstation after communication has been established with the central computer. The format of the SIGNON control statement is:

```
/*SIGNON NIHTTNRMTnnn password
```

NIHTTN must begin in column 10, RMTnnn in column 16, and password in column 73.

Parameters:

nnn	The remote terminal work station number.
password	The password is assigned by CIT. If the RJE operator wishes to change the password, the help desk must be notified and a new password will be assigned.

SIGNOFF

The SIGNOFF control statement should be used at an RJE work station to notify the central computer that all processing has been completed. Thirty seconds after issuing this command, the telephone will be disconnected. The format of the SIGNOFF control statement is:

```
/*SIGNOFF
```

2.14 Displaying Job Status with IOF

The Interactive Output Facility (IOF) provides a comprehensive interface to the JES2 subsystem. IOF provides full-screen terminal access and a completely programmable interface to:

- Batch jobs
- JES2 queues
- JES2 devices
- The system log
- The extended MCS console

- Initiators
- Running tasks, TSO users, and jobs
- Alternate machines in a sysplex

IOF has extensive facilities to review and manage batch jobs and sysout data sets. At a glance you can see if your job succeeded. IOF shows exactly where and why the job failed if a problem occurred. You can cancel, release, and modify your jobs with simple commands. You can browse your sysout data before it is printed. IOF lets you change your sysout data set characteristics, such as forms and class, by overtyping the desired field. For more information about using IOF, see the *IOF User's Guide*, which you can order through CIT Publications (see section 1.3).

1. From the CIT/Titan Primary Option Menu, enter: **IOF**

You can also enter just the letter "I." The IOF Option Menu is displayed.

```

----- IOF Option Menu -----
COMMAND ====>

Select an option.  To get a detailed option menu, follow the option with "?".

blank - Your jobs          G  - Output Groups          M  - Monitor Tasks
 I   - Input jobs          J  - ALL jobs menu           QT - Quick Trainer
 R   - Running jobs        H  - Jobs with HELD output  MAIL - TSO Transmit Mail
 O   - Output jobs         PR - Mainframe Printers    NEW  - NEW functions
 L   - Browse SYSLOG       P  - IOF Profile           HELP - Online HELP

JOBNAMES ====>
Enter 1 to 8 generic job names above

SCOPE      ====>          ALL, ME or another user's USERID
                                     Enter HELP to see all valid SCOPE values

DEST       ====>
                                     Enter 1 to 8 destinations above

SORT       ====> AGE      Enter HELP to see all valid SORT values
F1=HELP    F2=SPLIT     F3=END      F4=RETURN   F5=RFIND    F6=RCHANGE
F7=UP      F8=DOWN      F9=SWAP     F10=LEFT   F11=RIGHT   F12=RETRIEVE

```

Jobs can be looked at in IOF based on the USER field of the JOB statement. If USER is not specified, the userid of the TSO submitter is assumed on the job. If the job is submitted by another job, the userid will propagate from the first job to the second job.

2. To see a list of your jobs, press Enter.

2.14.1 IOF Job List Menu

The IOF Job List Menu gives you detailed status for all the jobs you submitted.

```

----- IOF Job List Menu -----( 10 )-----
COMMAND ==>                               SCROLL ==> CURSOR
----- Input Jobs -----
-----JOBNAME--JOBID--ACT-STAT-OWNER----C-POSIT-PRTY-SRVCLASS-QUALIFIER-----
_   1 PERBXTR  J005744          $WON    L          5  BATCH
_   2 LINBATCH J005484          $WON    L          5  BATCH
_   3 BHCXSLOD J005514          $WON    L          5  BATCH
----- Running Jobs -----
-----JOBNAME--JOBID--ACT-STAT-SYID-----CPU-----I/O---STEP-----PROCSTEP-SWP
_   4 $IGH     T005726          SYS4    :01        807  ADA     TCP40115 OUT
_   5 CCITCPGW S000538          SYS4    1:01       157  CCITCPGW CCITCPGW IN
_   6 SMTP     S000555          SYS4    5:04      168274 SMTP   SMTP   IN
_   7 $ZY6     T006181          SYS4    :01        703  ADA     V1E37920 OUT
_   8 $K6S     T006182          SYS4    :05        718  ADA     V1E36200 IN
----- Output Jobs -----
-----JOBNAME--JOBID--ACT-STAT-OWNER----DEST/DEVICE-----RECS-HELD-DAY--TIME
_   9 JVS2002  J006168          8 $WON   NIHJES2          859 070 10:21
_  10 $WONZ    J006169          CAN $WON   PRINTR30          35  070 10:21

F1=HELP      F2=SPLIT     F3=END        F4=RETURN     F5=RFIND      F6=RCHANGE
F7=UP        F8=DOWN      F9=SWAP       F10=LEFT     F11=RIGHT     F12=RETRIEVE

```

Some information is common to all job types:

- Line command area
- Menu number
- Jobname
- Jobid
- Action column (ACT) for command confirmation
- Job status column (STAT) that shows exception conditions

The IOF Job List Menu shows Input Jobs, Running Jobs, and Output Jobs. Each section shows the information that is most useful for that type of job. You can scroll RIGHT to get additional information about each job on the list.

You can select jobs for online review, cancel jobs, modify job characteristics such as class, destination and priority, and release held data sets.

Most commands can be entered in the line command area to the left of each job. Job characteristics can be modified by using the **Tab** key to place the cursor in the applicable field and overtyping the existing data.

Selecting a Job

To select a job, enter **S** in the line command area or enter its menu number in the command area at the top of the screen.

Other line commands are:

?	Display a function menu of available job commands. When you can't remember the commands, enter a question mark by the job.
B	Browse the entire job as one continuous data set
C	Cancel the job
R	Release all the held data sets of the job.
P	Print the job at your default print destination and sysout class.
ED	Recreate original input job from an output job and edit it. (ISPF only)
PC	Show why job is not printing. (TSO only)

2.14.2 IOF Job Summary Screen

Selecting a job produces the IOF Job Summary screen.

The IOF Job Summary screen has three sections:

- The **Job Information Section** at the top of the screen gives you basic information about your job.
- The **Step Information Section** in the middle contains one line for each job step in your job.
- The **Data Set Section** at the bottom has one line for each sysout data set the job produced.

You can see which steps completed normally with return code 0 (RC column), which abended, and which were not run.

All sysout data sets for the job are displayed in the data set section at the bottom. Both held and non-held data sets can be selected for browsing. Data set characteristics can be modified by overtyping. Additional data set characteristics can be seen by scrolling right or by requesting a data set function menu.

The IOF Job Summary screen provides commands for reviewing and modifying sysout data sets. You can browse sysout data, release individual held data sets, cancel

and modify held sysout data sets and output groups, and modify output characteristics such as destination and forms.

Data set line commands can be entered in the line command area to the left of each data set, and attributes can be changed by simply overtyping on the menu.

Selecting a Data Set

Enter **S** in the line command area to the left of a data set or enter its menu number in the command area.

Other line commands are:

?	Display a function menu for the data set. When you are in doubt or can't remember the available commands, enter a question mark for assistance.
C	Cancel a held data set.
M	Display a menu showing all data set attributes that can be modified.
R	Release a held data set.
P	Print an exact copy of the data set.
ED	Edit a temporary copy of the data set with the ISPF editor.

2.14.3 IOF Data Set Browse Screen

Selecting a sysout data set produces the IOF Data Set Browse screen.

On the IOF Data Set Browse screen, the top line shows the ddname, step name, procedure step name, current page number, line number, and columns.

The data set is displayed next. You can use the PF keys for UP, DOWN, LEFT, RIGHT and FIND to move within the data set.

Press F3 (END) to return to the IOF Job Summary screen, or F4 (RETURN) to return to the Job List Menu.

2.14.4 Saving job output to a data set

You can save all or part of your job output to a data set.

1. On the IOF Job List Menu, select the job whose output you want to save.

Enter the job number in the command area or enter **S** as a local command next to the job.

-
2. In the command area of the IOF Job Summary screen, enter: SD
SD stands for Snap Definition.
 3. On the IOF Snap Data Set Options screen, you can:
 - Press Enter to accept the default name for the target data set.
or
 - Change the name of the target data set and press Enter.
 - You might also want to change the SMS management class (MGMTCLAS).
 4. On the IOF Job Summary screen, snap the output.
 - To snap part of the output, type N as a local command next to each part to snap (LOG, JCL, MESSAGES, etc) and press Enter.
 - To snap all of the output, enter SNAP in the command area.

When you press Enter, SNP appears the Action column to show the data sets that were snapped.
 5. Keep pressing PF3 until you return to the IOF Option Menu to save the data set.

Alternative Methods

With each of these methods, you must name the target data set on the IOF Snap Data Set Options screen before you snap the output. After you have snapped all the output, you must keep pressing PF3 until you return to the IOF Option Menu to save it.

- On the IOF Job List Menu, browse the job whose output you want to save.

Enter the job number plus B in the command area (2B, for example) or enter B as a local command next to the job.

On the IOF Data Set Browse screen, enter: SNAP

All of the output will be saved to the target data set.

- On the IOF Job Summary screen, select the output.

On the IOF Data Set Browse screen, enter: SNAP

The selected output will be saved to the target data set.

2.15 Displaying Job Status with UDF

The User Display Facility (UDF) allows users to display information about jobs processed by ThruPut Manager. If ThruPut Manager is delaying the execution of your job, UDF will allow you to display information such as hold reasons, job dependencies, and needed data sets.

UDF is invoked through the ISPF command TMUSER. You can assign TMUSER to a PF key. Follow the instructions below to use UDF:

1. Invoke UDF in IOF using one of two ways:
 - Place the cursor at the job name and press the PF key you have assigned to TMUSER.
 - or
 - Enter the TMUSER command on the ISPF command line, move the cursor to the job name, and press Enter.

You will see one or more Information Summary Lines. For example:

```
GL3005TB(J01143) _ DC JB JC JL      H
```

The Information Summary Line contains the job name and job number. It may also contain a command line and one or more acronyms indicating the ThruPut Manager services affecting the job. The most common acronyms are:

DC	Dataset Contention Services
JB	Job Binding Services
JC	Job Chaining Services
JL	Job Limiting Services
JS	Job Setup Services
MH	Multi-Hold Services
RS	Robotic Setup Services

Highlighted acronyms indicate that the service is causing the job to be held.

The line may also contain the letters **H** or **D**, showing whether the job has been held or deferred by ThruPut Manager.

2. To see more information about an acronym, place the cursor under the acronym and press Enter.
3. You can enter the following commands on the command line:
 - **D** to see job details

-
- **V** to see which volumes are used

2.15.1 Resolving Data Set Contention

If the information summary line contains a DC acronym (data set contention), place the cursor on DC and press Enter. You will see the names of the data sets the job is waiting for. Place the cursor on a data set name and press Enter to display the name of the job that is holding the data set.

2.16 Displaying and Purging Jobs through the Web

The Display or Purge Jobs facility on the Web enables you to locate jobs and fetch output. You can also use it to purge an output job.

To use the facility, go to:

<https://titan.nih.gov/util/dispjoc>

2.17 Avoiding Job Turnaround Delays

Users who are accustomed to the turnaround times at the NIH Computer Center get concerned when turnaround time increases and wonder about the cause. Delays can be caused by equipment malfunction or system problems. While those delays are unpredictable, there are other delays that can be understood and even avoided.

A batch job goes through several stages before it begins to execute the programs in the job. In the first stage, awaiting execution, it shows up in IOF as an input job and is simply waiting for the operating system to select the job for execution. Very few system resources are used during this stage.

Before the job begins execution, ThruPut Manager determines if the necessary resources are available (specified by `/**+JBS BIND` statements) and if job serialization dependencies have been met (specified by `/*BEFORE`, `/*AFTER`, `/*WITH`, or `/*CNTL` statements). Multiple jobs with the same job name and the same userid cannot execute simultaneously.

Once these conditions have been met, ThruPut Manager determines which data sets are needed and the type of control (shared or exclusive) needed for each data set. If a needed data set is not available, ThruPut Manager sends a DTM7110I message to the JES2 JOB LOG of the job indicating the data sets needed. If the needed data set is allocated by another TSO user, ThruPut Manager sends the TSO user messages DTM7108I and DTM7109I asking the user to free the data set and specifying the job that needs it.

You can use the User Display Facility (see section 2.3.3) to display information about jobs being processed by ThruPut Manager. If the execution of your job is delayed, you can display hold reasons, job dependencies, and the names of needed data sets.

After all the data sets are available, the job begins executing and shows up in IOF as a running job.

2.18 Understanding Job Output

This section gives an elementary explanation designed to help beginners find their way through the output of a job. It is followed by a sample job output. There are six sections to the output of any job:

Header sheet (not illustrated) For jobs printed at the central facility, this has CIT's address. The number of the user's outbox is shown in large numerals. The job name, the job number, the date and time the job was printed, the printer number, and the userid are listed below the CIT logo.

When the job is printed at a remote printer the following information appears: box number, job number, job name, date and time job was printed, programmer's name, printer number and CIT's address.

JES2 job log The JES2 job log includes the **expanded JCL** and the **operating system messages** if you use the NIH default for MSGLEVEL.

To the left of each line on this page is a 24-hour clock time (of the pattern: hour.minute.second). The ICH70001I message shows the userid under which this job was run.

The \$HASP373 message shows the time the job started execution, the job number, the job name, the initiator used, the job class, subsystem it ran on, and the date it was run.

The lines that begin with a dash (-) show the return code, the I/O count (one for each block of data read or written), the CPU time, and the estimated cost of each step in the job.

The final line is the \$HASP395 message which shows the time the job ended. There may be other system messages.

If there were an error in the job, the job log would include error messages indicating an ABEND, JCL error, or hardware or software failure.

Expanded JCL This page begins with the JOB statement and job number. It lists both the JCL statements submitted by the user (preceded by //) and the JCL statements from invoked procedures (preceded by XX) as

well as the messages which show the symbolic substitutions performed. Any statements in procedures that are overridden by user-supplied statements are preceded by X/. The JCL statements are numbered to the left so they can be referred to by later messages.

Operating system messages

This section contains the messages generated by the system for each job step. These messages give such information as unit allocations and deallocations, data set dispositions, volume serial numbers, and step completion codes. Any JCL errors are also noted in the step where the error occurred. The number to the left of each message is the number of the JCL statement it refers to in the expanded JCL listing.

At the end of the Operating system information come the job deallocation messages issued when the system cleans up data sets that are still allocated at the end of the job.

Program output

The contents of this section vary depending on the nature of the job. They might include: compiler output, Binder output, reports generated by the program, messages indicating whether the job ran successfully or not, and messages from utility programs. Since the sample is a simple copy, it does not include the compiler output and Binder output that would have been present if a language such as FORTRAN had been used.

Trailer sheet

This is similar to the cover sheet; the end time for the job indicates when the job was printed.

Sample Job Output

```

1          J E S 2  J O B  L O G  --  S Y S T E M  4 0 9 0  --  N O D E  N I H J E S 2
0
14.26.30 JOB19251 ---- WEDNESDAY, 20 MAR 2002 ----
14.26.30 JOB19251 IRR010I USERID $III      IS ASSIGNED TO THIS JOB.
14.26.31 JOB19251 ICH70001I $III        LAST ACCESS AT 14:23:44 ON WEDNESDAY, MARCH 20, 2002
14.26.31 JOB19251 $HASP373 JDASAMPL STARTED - INIT 7      - CLASS L - SYS 4090 - 20 Mar 2002
14.26.31 JOB19251 IEF403I JDASAMPL - STARTED - TIME=14.26.31
14.26.31 JOB19251 -JOBNAME  STEPNAME  PROCSTEP   RC   I/O COUNT   CPU TIME(SECONDS)   ESTIMATED
COST
14.26.31 JOB19251 -JDASAMPL DSSCR    SCRSTEP    00           13           .03
$.05
14.26.31 JOB19251 -JDASAMPL COPY    COPYSTEP    00           21           .04
$.07
14.26.31 JOB19251 IEF404I JDASAMPL - ENDED - TIME=14.26.31
14.26.31 JOB19251 -JDASAMPL ENDED.  JOB I/O COUNT=    34  JOB CPU TIME=    .07  EST. COST=    $.14
14.26.32 JOB19251 $HASP395 JDASAMPL ENDED
0----- JES2 JOB STATISTICS -----
-   20 MAR 2002 JOB EXECUTION DATE
-           11 CARDS READ
-           116 SYSOUT PRINT RECORDS
-           0 SYSOUT PUNCH RECORDS
-           6 SYSOUT SPOOL KBYTES
-           0.01 MINUTES EXECUTION TIME
1
1 //JDASAMPL JOB  (), 'CITUSER',MSGCLASS=H,NOTIFY=$III                JOB19251
//*
//*  SAMPLE JOB FOR TITAN BATCH MANUAL
//*
2 //SCRSTEP      EXEC      DSSCR,
//              NAME=$III.NEW.COPY.DATA
XX*
XX*      SCRATCHES A DATA SET STORED ON DISK
XX*
XX*      SYS - 04/20/87
XX*
3 XX PROC  NAME=,                (REQ)  DATA SET TO BE SCRATCHED
XX        STORAGE=FILE,         DISK UNIT
XX        DISK=                  DISK VOLUME SERIAL
XX*
4 XXDSSCR EXEC PGM=DSSCR,REGION=512K,PARM='VOL="&DISK" ,UNIT="&STORAGE" '
XX*
IEFC653I SUBSTITUTION JCL - PGM=DSSCR,REGION=512K,PARM='VOL=" " ,UNIT="FILE" '
5 XXDSNAME DD  DSNAME=&NAME,
XX          VOL=REF=NIH.EMPTY,DISP=OLD
IEFC653I SUBSTITUTION JCL - DSNAME=$III.NEW.COPY.DATA,VOL=REF=NIH.EMPTY,DISP=OLD
6 XXSYSPRINT DD SYSOUT=*
7 XXSYSUDUMP DD SYSOUT=*
//*
8 //COPYSTEP     EXEC      COPY
XX*
XX*      COPIES A SEQUENTIAL DATA SET
XX*
XX*      CIA - 02/02/01
XX*
9 XXCOPY        EXEC PGM=ICEGENER
XX*
10 XXSYSPRINT DD  SYSOUT=*
11 //COPY.SYSUT1 DD          DSN=AAAJDA.WEEKDAY.LIST,DISP=SHR
X/SYSUT1 DD  DISP=SHR
12 //COPY.SYSUT2 DD          DSN=$III.NEW.COPY.DATA,UNIT=TMP,
//              SPACE=(TRK,(5,1),RLSE),DISP=(NEW,CATLG)
X/SYSUT2 DD  DISP=(NEW,KEEP,DELETE)
13 XXSYSIN      DD  DUMMY
1
STMT NO. MESSAGE
2 IEFC001I PROCEDURE DSSCR WAS EXPANDED USING SYSTEM LIBRARY NIH.PROCLIB
8 IEFC001I PROCEDURE COPY WAS EXPANDED USING SYSTEM LIBRARY NIH.PROCLIB

```

```

DTM1459I 2002.079 14:26:31 JOB JDASAMPL REQUEUED TO CLASS=L
IEF236I ALLOC. FOR JDASAMPL DSSCR SCRSTEP
IEF237I 0320 ALLOCATED TO DSNAME
IEF237I JES2 ALLOCATED TO SYSPRINT
IEF237I JES2 ALLOCATED TO SYSUDUMP
IEF237I 0C9E ALLOCATED TO SYS00001
IEF142I JDASAMPL DSSCR SCRSTEP - STEP WAS EXECUTED - COND CODE 0000
IEF285I $III.NEW.COPY.DATA KEPT
IEF285I VOL SER NOS= NIHLBD.
IEF285I $III.JDASAMPL.JOB19251.D0000101.? SYSOUT
IEF285I $III.JDASAMPL.JOB19251.D0000102.? SYSOUT
IEF285I SYS02079.T142631.RA000.JDASAMPL.R0147994 KEPT
IEF285I VOL SER NOS= DSP126.
IEF373I STEP/DSSCR /START 2002079.1426
IEF374I STEP/DSSCR /STOP 2002079.1426 CPU OMIN 00.03SEC SRB OMIN 00.00SEC VIRT 48K SYS
IEF236I ALLOC. FOR JDASAMPL COPY COPYSTEP
IEF237I JES2 ALLOCATED TO SYSPRINT
IGD103I SMS ALLOCATED TO DDNAME SYSUT1
IGD101I SMS ALLOCATED TO DDNAME (SYSUT2 )
          DSN ($III.NEW.COPY.DATA )
          STORCLAS (STANDARD) MGMTCLAS (STANDARD) DATACLAS ( )
          VOL SER NOS= DSP124
IEF237I DMY ALLOCATED TO SYSIN
IEF142I JDASAMPL COPY COPYSTEP - STEP WAS EXECUTED - COND CODE 0000
IEF285I $III.JDASAMPL.JOB19251.D0000103.? SYSOUT
IGD104I AAAAJDA.WEEKDAY.LIST RETAINED, DDNAME=SYSUT1
IGD104I $III.NEW.COPY.DATA RETAINED, DDNAME=SYSUT2
IEF373I STEP/COPY /START 2002079.1426
IEF374I STEP/COPY /STOP 2002079.1426 CPU OMIN 00.04SEC SRB OMIN 00.00SEC VIRT 4116K SYS 276K
IEF375I JOB/JDASAMPL/START 2002079.1426
IEF376I JOB/JDASAMPL/STOP 2002079.1426 CPU OMIN 00.07SEC SRB OMIN 00.00SEC
1
  DSSCR11 - SCRATCH SUCCESSFUL FOR $III.NEW.COPY.DATA ON DSP126
  DSSCR12 - UNCATALOG SUCCESSFUL FOR $III.NEW.COPY.DATA
1ICE143I 0 BLOCKSET COPY TECHNIQUE SELECTED
ICE000I 0 - CONTROL STATEMENTS FOR 5740-SM1, DFSORT REL 14.0 - 14:26 ON WED MAR 20, 2002 -
0 OPTION COPY,MSGDDN=SYSPRINT,SORTIN=SYSUT1,SORTOUT=SYSUT2
ICE201I 0 RECORD TYPE IS F - DATA STARTS IN POSITION 1
ICE193I 0 ICEAM2 ENVIRONMENT IN EFFECT - ICEAM2 INSTALLATION MODULE SELECTED
ICE088I 0 JDASAMPL.COPYSTEP.COPY , INPUT LRECL = 80, BLKSIZE = 11440, TYPE = FB
ICE093I 0 MAIN STORAGE = (MAX,4194304,4194304)
ICE156I 0 MAIN STORAGE ABOVE 16MB = (4136944,4136944)
ICE127I 0 OPTIONS: OVFL0=RC0 ,PAD=RC0 ,TRUNC=RC0 ,SPANINC=RC16,VLS0MP=N,SZERO=Y,RESET=Y,VSAM0MT=Y
ICE128I 0 OPTIONS: SIZE=4194304,MAXLIM=1048576,MINLIM=450560,EQUALS=N,LIST=Y,ERET=RC16
,MSGDDN=SYSPRINT
ICE129I 0 OPTIONS: VIO=N,RES0NT=ALL ,SMF=NO,WRKSEC=Y,OUTSEC=Y,VERIFY=N,CHALT=N,DYNALOC=N,ABCODE=MSG
ICE130I 0 OPTIONS: RESALL=4096,RESINV=0,SVC=109
,CHECK=Y,WRKREL=Y,OUTREL=Y,CKPT=N,STIMER=Y,COBEXIT=COB1
ICE131I 0 OPTIONS: TMAXLIM=4194304,ARESALL=0,ARESINV=0,OVERRGN=16384,CINV=Y,CFW=Y,DSA=0
ICE132I 0 OPTIONS: VLSHRT=N,ZDPRINT=N,IEXIT=N,TEXTIT=N,LISTX=N,EFS=NONE,EXITCK=S,PARMDDN=DFSPARM
ICE133I 0 OPTIONS: HIPRMAX=OPTIMAL,DSPSIZE=MAX ,ODMAXBF=0,SOLRF=Y,VLLONG=N,VSAMIO=N
ICE084I 0 EXCP ACCESS METHOD USED FOR SYSUT2
ICE084I 0 EXCP ACCESS METHOD USED FOR SYSUT1
ICE090I 0 OUTPUT LRECL = 80, BLKSIZE = 27920, TYPE = FB (SDB)
ICE055I 0 INSERT 0, DELETE 0
ICE054I 0 RECORDS - IN: 82, OUT: 82
ICE052I 0 END OF DFSORT

```

3 Utilities - General Information

This section describes some of the utility programs available at the NIH Computer Center. These include IBM utilities, utility programs from other installations, and CIT-written utilities. Cataloged procedures have been set up for most of these programs to make certain common requests easier to perform. The description of each cataloged procedure or program includes instructions on its use and some of the functions available. These descriptions are not meant to replace the appropriate manuals when they are available.

3.1 IBM Utilities

IDCAMS	Creates a generation data group and defines the base index to the operating system.
IEBCOPY	Copies or merges partitioned data sets, selects or excludes specified members in a copy operation, and renames or replaces selected members of a partitioned data set.
IEBDG	Creates test data sets consisting of patterns of data. This utility can be used to read and write partitioned and sequential data sets.
IEBGENER	Copies records from a sequential data set or a member of a partitioned data set and creates either a sequential or partitioned data set.
IEBPTPCH	Prints or punches a data set. Not supported by Titan for local output.
IEBUPDTE	Updates sequential or partitioned data sets.
IEHPROGM	Scratches data sets or members, renames data sets, and catalogs and uncatalogs data sets.

For further information concerning batch IBM utility programs, contact the NIH Help Desk or consult the manual *IBM DFSMS/MVS Utilities*, SC26-4926

For information on the transition to Titan from the former North and South systems, go to:

<http://silk.nih.gov/silk/titan>

The following procedures were available on the South system but are not available on Titan: DISKEXPT, DSSCM, DSSCMI, DSUNSCM, DSUNSCMI, MEMALTER, MEMRENAM, PALTAPE, PASBILL, PDSMAP, POSTSCRIP, PTFORM, RESPOUT, TAPECOPY, TAPESCAN, VOLSTAT.

The following procedures were available on the North system but are not available on Titan: DYNAMO II, GPSS, FRGNCOPY, FRGNOUT, FRGNOUTC, ORACLE, PCCDCODE, PCCNCODE, PCCVTOC, SANITIZE, SYNCSORT.

3.2 Restrictions at NIH

In order to prevent accidental modification to the system and to safeguard customer data, there are restrictions on the use of some utility programs.

Use of the following IBM system utilities is forbidden:

IEHATLAS	IEBIMAGE
DSF	IFHSTATR
IEHINITT	ICAPRTBL

The following IBM system utilities and service aids should not be used to access any of the public DASD volumes. There are NIH Computer Center utilities to provide each of these functions.

AMASPZAP	(use MEMALTER)
IEHMOVE	(use PDSSAVE, PDSGET, DDSSAVE, DDSGET)
IEHPROGM	(use CATDS, UNCATDS)
IEHLIST	(use PDSMAP)

All backup and recovery of direct access volumes must be performed through facilities provided by the NIH Computer Center. The IBM dump/restore utility, DF/DSS, may not be invoked directly. Use the NIH Computer Center utilities DISKSAVE and DISKGET.

These IBM data set utilities may be used to access data on public and private volumes:

IEBCOMPR	IEBGENER
IEBCOPY	IEBPTPCH
IEBDG	IEBUPDTE
IEBEDIT	

While the NIH Computer Center has no restrictions on dynamically invoking the utilities, it supports only the interfaces published in the NIH Computer Center documentation. Users should do their best to analyze and isolate problems that develop in the applications invoking these utilities before seeking assistance in problem diagnosis and resolution.

Users should not develop software that depends on particular output formats from NIH Computer Center utilities. SYSOUT changes are not always announced in advance.

Please note: Formal action will be taken against anyone who attempts to circumvent protection software and threatens the account and data security of the system.

4 Data Set Utilities

4.1 The Data Set Wonder Utility

4.1.1 EDSUTIL

What it can do

EDSUTIL is a one-stop wonder utility that can handle edit format data sets as well as non-edit formats. Edit format is the standard format used by WYLBUR to store data sets (see section 4.13). Non-edit format refers to all other formats. EDSUTIL can copy, extract portions of, and change the DCB characteristics of data sets. The many capabilities of EDSUTIL and the complete specifics of EDSUTIL syntax are described in the *Edit Format Utility Package* manual. EDSUTIL is an NIH-written utility.

The EDSUTIL program can be used to copy or list sequential data sets. Edit format data sets may be converted to non-edit format and non-edit format data sets may be converted to edit format. DD statements are used to define the data sets to be copied or listed. Members of a partitioned data set may be processed by specifying the member name on the DD statement. Multiple copying operations may be used to combine multiple data sets and copy to one or more data sets. The data sets being copied or the data sets created by the copying mechanism may be listed. There are options provided to permit the user to describe the method used for handling line numbers. Also, the user may call programs to be executed, and upon completion of execution of each program, return control to the EDSUTIL program.

The main purpose of the EDSUTIL program is to convert data sets between edit and non-edit format. The program also allows copying data sets from edit to edit format, or from non-edit to non-edit format. The EDSUTIL program may be used to change the record format when copying non-edit format data sets. In general, however, it may be more efficient to use programs other than EDSUTIL when not converting between edit and non-edit format. This is because EDSUTIL handles records on a logical record basis, which is not as efficient as a program that handles records on a physical block basis.

Each data set referenced by a command must have a DD statement supplied by the user. The ddname specified for a data set on the DD statement is used in the commands to indicate the data set to be processed. If a data set to be accessed by EDSUTIL has the DCB attributes of RECFM=U and BLKSIZE greater than or equal to 1022, it is assumed to be an edit format data set. In the case of writing a data set, if no DCB parameters are specified on the associated DD statement and the data set is not directed to a SYSOUT class, the data set is written in edit format.

Syntax of Commands

Comments may be used in commands by beginning the comment with a percent sign (%). The comment is terminated by the end of the line. A blank or empty line is

treated as a comment line. More than one command may be included in a line by separating the commands with a semicolon (;). A command may be continued by including two percent signs (%%) at the location where the command is being split. The next line is used as the continuation line unless it is a blank line or a comment line. Any information that is on the same line and follows the two percent signs (%%) is treated as a comment. The percent signs must appear after a complete option; that is, if an option of a command consists of more than one word, the percents must follow the last word in the option. The abbreviations used in WYLBUR may also be used to abbreviate any of the commands or options for the EDSUTIL program.

EDSUTIL COPY Command

The EDSUTIL COPY command is used to copy a data set to another data set. Conversion to and from edit format is supported. A data set is designated by specifying the ddname associated with the DD statement that defines the data set. The general format of the COPY command is:

```
COPY DDNAME=input-ddname TO DDNAME=output-ddname
```

where "input-ddname" is the ddname of the DD statement defining the data set to be copied and "output-ddname" is the ddname of the DD statement defining the data set to be written. For example, to copy a data set with a ddname of INPUT, to a data set with the ddname OUTPUT, the following command would be used:

```
COPY DDNAME=INPUT TO DDNAME=OUTPUT
```

More than one data set may be copied by including the ddnames of the data sets, separated by the word AND, following COPY and preceding the TO portion of the command.

In the following example, the COPY command instructs the EDSUTIL program to copy the contents of the data set with a ddname of IN1, and then the contents of the data set with a ddname of IN2, to a data set with the ddname OUTPUT.

```
COPY DDNAME=IN1 AND DDNAME=IN2 TO DDNAME=OUTPUT
```

To copy data sets to more than one output data set, the data sets to be written are specified by including the ddnames of the output data sets, separated by AND, following the word TO. For example, to copy a data set with the ddname INPUT to a data set with the ddname OUT1 and to a data set with the ddname OUT2, the following command could be used:

```
COPY DDNAME=INPUT TO DDNAME=OUT1 AND DDNAME=OUT2
```

There are a number of options available to indicate special handling of the data sets. These options must be specified after the ddname of each applicable data set. The

options may be specified in any order. If an option is repeated or if two incompatible options are coded, then only the last one encountered is used.

LIST The contents of the data set are to be listed. Only the input or the output can be listed; input and output data sets cannot both be listed.

The listing is directed to the data set specified by the SYSPRINT DD statement.

If line numbers are part of the data set being listed, they are included in the listing. The line number appears to the left of the text of the line and is separated from the text by two blanks.

A number of options are available to specify how the listing is to be performed. These options, described in the LIST command (see below), are:

CC
MC
INDENT n
DOUBLE
TRIPLE
SPACING n
MARKER c

SKIP integer Specifies the number of lines to be skipped before a line is read or written.

TAKE integer Specifies the number of lines to be read or written.

RECFM U Indicates that the data set has edit format characteristics (RECFM is U and BLKSIZE is greater than or equal to 1022), but the data set is not to be treated as an edit format data set.

The following examples illustrate some uses of these options:

```
COPY DDNAME=INPUT TAKE 100 TO DDNAME=OUTPUT
```

The first 100 records of the data set with a ddname INPUT are copied to a data set with the ddname OUTPUT.

```
COPY DDNAME=IN1 LIST AND DDNAME=IN2 LIST TO DDNAME=OUTPUT
```

The contents of data sets with ddnames IN1 and IN2 are listed to the data set designated by the SYSPRINT DD statement. The data set being written, with a

ddname OUTPUT, contains the records from the data set with the ddname IN1 followed by the records from the data set with the ddname IN2.

EDSUTIL LIST Command

The EDSUTIL LIST command is used to obtain a listing of a data set. The listing is directed to the data set specified by the SYSPRINT DD statement. If line numbers are being listed with the data set, the line number is separated from the text of the line by two blanks. A data set is designated by specifying the ddname associated with the DD statement, which defines the data set following the word LIST. For example, to list the contents of the data set with a ddname of DATA, the following command would be used:

```
LIST DDNAME=DATA
```

More than one data set may be listed by including the ddnames of the data sets, separated by AND, following the word LIST. The data sets are listed in the order in which they appear in the command. For example, to list the data sets with ddnames PROGRAM and DATA, the following command could be used.

```
LIST DDNAME=PROGRAM AND DDNAME=DATA
```

The TO DDNAME=ddname option may be used to direct the listing to a specified data set. For example, the following command could be used to list the data set with a ddname of INPUT to the data set defined by a DD statement LISTING:

```
LIST DDNAME=INPUT TO DDNAME=LISTING
```

To direct the listing to more than one data set, the data sets to receive the listing are specified by including the ddnames of these data sets, separated by AND, following the word TO. For example, to list the data set with a ddname of INPUT to a data set with the ddname of LIST1 and to a data set with the ddname of LIST2, the following command could be used:

```
LIST DDNAME=INPUT TO DDNAME=LIST1 AND DDNAME=LIST2
```

There are a number of additional options available that may be specified for each output data set referenced to indicate instructions for how the data set listings are to be done. These options must be specified after the ddname of each applicable data set.

CC The data set is processed assuming it contains ISO/ANSI carriage control characters in the first column of each line. For more information on ISO/ANSI carriage control, see the *Titan User's Guide*.

INDENT n Indicates the number of blanks to be inserted before the text of each line is listed. If line numbers are being listed with the data

set, the blanks are inserted before the line number.

- DOUBLE** The output is to be double spaced (i.e., one blank line before each line from the data set). To obtain this result, ISO/ANSI carriage control characters are inserted in each line. For more information on ISO/ANSI carriage control, see the *Titan User's Guide*.
- TRIPLE** The output is to be triple spaced (i.e., two blank lines before each line from the data set). To obtain this result, ISO/ANSI carriage control characters are inserted before each line. For more information on ISO/ANSI carriage control, see the *Titan User's Guide*.
- SPACING n** SPACING n, where "n" is a positive integer, indicates that n-1 blank lines should appear before each line in the data set. To obtain this result, ISO/ANSI carriage control characters are used. For more information on ISO/ANSI carriage control, see the *Titan User's Guide*.
- MARKER c** MARKER c, where "c" is a character, indicates that an eject to a new page should be performed if a line is encountered that only contains the character "c" in the first column. A line that contains a "c" in the first column and other characters in the line is listed. To perform the functions provided by this option, ISO/ANSI carriage control characters are inserted before each line. The marker character may be enclosed in either single or double quotation marks. For more information on ISO/ANSI carriage control, see the *Titan User's Guide*.

The following options, described above for the COPY command, may be used to specify the characteristics of the input data sets:

SKIP integer
TAKE integer
RECFM U

EDSUTIL Procedure

The EDSUTIL procedure is defined to provide some of the job control language necessary to execute the EDSUTIL program. The following symbolic parameter is available in the EDSUTIL procedure provides the first command to be executed. The symbolic parameter is optional.

Symbolic Parameter	Function
COMMAND	To specify the first command to be executed before the commands in the SYSIN data set, use

COMMAND='command'.

The default is to assume that the SYSIN data set contains all of the commands.

Examples

To obtain a listing of a cataloged edit format data set named userid.TEXT, the following job could be executed. The job uses the EDSUTIL procedure and the symbolic parameter COMMAND to specify the command. A DD statement with the name TEXT defines the data set to be listed.

```
//          EXEC EDSUTIL,COMMAND='LIST DDNAME=TEXT'  
//TEXT     DD   DSNAME=userid.TEXT,DISP=SHR
```

The following example copies an edit format data set to a non-edit format data set and produces a listing of the copied data set. The procedure EDSUTIL is used. The ddname of the data set to be copied is IN and the ddname of the data set to be written is OUT.

```
//          EXEC EDSUTIL  
//IN       DD   DSNAME=userid.IN,DISP=SHR  
//OUT      DD   DSNAME=userid.OUT,UNIT=TEMP,  
//          DISP=(NEW,CATLG),  
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=11440),  
//          SPACE=(TRK,(10,5),RLSE)  
COPY DDNAME=IN TO DDNAME=OUT LIST
```

In the previous example, the COPY command is the only command given to the EDSUTIL program. The command is specified in the SYSIN input data set (the system generates a //SYSIN DD * statement when one is not present). The same result could have been obtained by using the COMMAND symbolic parameter to specify the command instead of using the input data set.

4.2 Backing Up and Recovering Data Sets

This section describes the DSSAVE and DSGET procedures.

Backup and recovery can also be done using EDSSAVE (see section 4.13.5), EDSGET (see section 4.13.6), PDSSAVE (see section 4.15.8), PDSGET (see section 4.15.9), DDSSAVE (see section 4.2.1), and DDSGET (see section 4.2.2).

4.2.1 DSSAVE

DSSAVE saves a sequential data set from disk to tape by copying the data set to tape. DSSAVE executes the NIH-written utility DSCOPYJ.

Required Symbolic Parameters

NAME	Names the input data set; also names the output data set unless NEWNAME is specified.
RETPD	Specifies the number of days the data set on the output tape is to be kept.

Optional Symbolic Parameters

DISK	Supplies the input volume serial number. Required if the data set is not cataloged.
TAPE	Supplies the output volume serial number if a specific volser is required.
NEWNAME	Names the output data set.
STORAGE	Provides a unit name for input. Required if the data set is not cataloged.
STATUS	Specifies whether the output data set is OLD or NEW; NEW is the default.
DRIVE	Gives a unit name for output; the default is TAPE.
DEN	Specifies a code (3 or 4) to indicate the density for a 9-track tape.
SEQ	Gives the sequence number for the tape data set; the default is 1. Overwriting data set 1 destroys any subsequent data sets.
LBL	Defines the type of label on the output tape. SL (standard-labeled) is the default. NL (non-labeled) may be specified.
STEPEND	Indicates the final (end of step) disposition for the output data set; KEEP is the default.

The following example saves a cataloged data set from disk to tape. The data set resides on a disk volume and is being saved to a virtual tape that will be kept for 30 days.

Example

```
//stepname EXEC DSSAVE,NAME='userid.dsname',
// RETPD=30
```

The following example saves a cataloged data set from a disk volume to a foreign 3480 tape, makes it the third data set on the tape, and gives it a new name. Notice that

the DD statement for the output tape must be overridden to specify EXPDT=98000 since it is a foreign tape.

Example

```
//stepname EXEC DSSAVE,NAME='userid.dsname1',  
// DRIVE=FRGN3480,TAPE=tapeser,SEQ=3,  
// NEWNAME='userid.dsname2'  
//OUT DD EXPDT=98000
```

4.2.2 DSGET

DSGET copies a sequential data set from tape to disk by copying the data set to disk. It executes the NIH-written utility DSCOPYJ.

Required Symbolic Parameters

NAME Names the input data set; also names the output data set unless NEWNAME is specified.

Optional Symbolic Parameters

NEWNAME Names the output data set.

TAPE Supplies the input volume serial number. Required if the data set is not cataloged.

STORAGE Gives a unit name for output; the default is LONGTERM.

STATUS Specifies whether the output data set is OLD or NEW; NEW is the default.

DRIVE Gives a unit name for input; the default is TAPE.

DEN Specifies a code (3 or 4) to indicate the density for a 9-track tape.

SEQ Gives the sequence number for the tape data set; the default is 1.

UNITS Gives the units of allocation for the SPACE parameter; the default is TRK (tracks).

SIZE Gives the primary space allocation for the disk data set; the default is 10.

INCR Gives the number of units in each secondary allocation; the default is 5.

UNUSED Determines whether unused space is released; the default is RLSE.

Code UNUSED= to retain unused space.

LBL	Defines the type of label on the tape. SL (standard-labeled) is the default. NL (non-labeled) may be specified, but DCB information must be supplied if the input tape is NL.
LRECL	Gives the logical record length for the input data set; it is required if LBL=NL and should not be used for standard-labeled tapes.
BLKSIZE	Gives the block size for the input data set; it is required if LBL=NL and should not be used for standard-labeled tapes.
RECFM	Gives the record format for the input data set; it is required if LBL=NL and should not be used for standard-labeled tapes.
STEPEND	Indicates the final (end of step) disposition for the output data set; CATLG is the default.
DISK	Supplies the output volume serial number. Required if the data set is to be written to a dedicated disk.

The following example copies a data set from a cartridge tape to a cataloged data set with a new name in the LONGTERM management class.

Example

```
//stepname EXEC DSGET,NAME='userid.dsname1',  
// TAPE=tapeser,NEWNAME='userid.dsname2'
```

The following example copies a data set from a cartridge tape to a cataloged data set in the NOBACKUP management class, changing the default space allocations.

Example

```
//stepname EXEC DSGET,NAME='userid.dsname',  
// TAPE=tapeser,  
// STORAGE=NOBACKUP,SIZE=primary,  
// INCR=secondary,UNITS=CYL
```

The following example copies the second data set from an unlabeled foreign 3490 cartridge tape to a LONGTERM data set, renaming the data set and cataloging it.

Example

```
//stepname EXEC DSGET,NAME='dsname1',  
// TAPE=tapeser,LBL=NL,  
// LRECL=80,BLKSIZE=4000,RECFM=FB,SEQ=2,DRIVE=FRGN3490,
```

```
// NEWNAME='userid.dsname2'
```

4.2.3 Hints on Backing Up and Recovering Data Sets

- When exporting an edit format data set, use the EDSSAVE procedure. This procedure converts edit format data sets to a format that can be read at other installations. When importing a data set that you wish to access through WYLBUR, use the EDSGET procedure. This procedure converts non-edit format data sets to edit format.
- It is advisable to maintain backups of important and volatile disk data sets rather than relying entirely on the NIH Computer Center's backup facilities.
- Because of the problems that may occur with multi-data set tapes, CIT recommends that no more than 100 data sets be saved per tape.
 - When changing a data set on a multi-data set tape, all information beyond the point of modification is destroyed. For example, if a tape has 100 data sets and data set 10 is replaced, data sets 11 through 100 are destroyed.
 - If you are creating a multi-data set tape and have forgotten how many data sets are already on the tape, don't guess. You could destroy important data. Instead, use the TAPEINFO procedure. This procedure gives the data set name and other attributes for each file on the tape.
 - Be sure to check output before adding more data sets when creating a multi-data set tape. The data sets must be written in direct sequence. If, for example, a previous job to write data set 8 was unsuccessful, the current job to write data set 9 will fail.
- The normal return code from IEHMOVE (used by PDSSAVE and DDSSAVE) is 0. Anything higher should be scrutinized to determine whether the copy was performed.
- If the wrong procedure is used (e.g., DDSSAVE for a partitioned data set), the return code will be 8, and the data set will not be copied.
- These procedures cannot process data sets where the DSN has a qualifier of more than 8 characters. In this case an error message is issued.

4.3 Cataloging Data Sets

The system catalog is a data set of information maintained by the operating system that is used to keep track of the location of data sets. Entries in the catalog include the name of the data set, the volume on which the data set is stored, and the type of device on which the data set is stored (e.g., a 3390 disk or a magnetic tape).

Any program can reference a cataloged data set (i.e., a data set whose name is in the catalog) by giving only its name. The system looks at the catalog to determine where the data set is stored, so the user doesn't need to specify the volume on which the data set is stored. This can be particularly useful when sharing a data set with other people. If other users are using the catalog to access the data set, you can move the data set without informing the other users of the new location.

There are restrictions on the names of data sets:

- Data sets must follow the Titan naming convention of beginning with either a userid (userid.dsname) or an account group (account.dsname).
- IBM standard naming conventions must also be followed. That is, periods must be used to divide the data set name into groups of eight or fewer characters, and each group of eight or fewer characters must begin with an alphabetic or national (@, \$, #) character. The total length including required characters, optional characters, and all periods cannot exceed 44.
- Only one data set with a given name can be cataloged.

Most data sets are cataloged or uncataloged by means of disposition processing, ISPF commands, or TSO commands. The CATDS procedure can be used for all data sets. UNCATDS can be used for data sets stored on tapes.

The ISPF panel 3.4 and the TSO command LISTCAT can be used to obtain information about cataloged data sets.

The DSNNAME and DISP parameters are always required to access a cataloged data set. The LABEL parameter is required only if the volume the data set resides on does not have a standard label. DCB parameters are required only if the processing program or the label does not supply full DCB information. Do not supply volume serial numbers or device types when accessing data sets through the catalog.

4.3.1 VDSUTIL

The VDSUTIL procedure scratches or renames multiple cataloged data sets in a single step using the IBM program IDCAMS. This provides an alternative to the DSSCR procedure which can scratch only a single data set at a time. The management class for multiple cataloged data sets can also be changed by using VDSUTIL. In addition, VDSUTIL can be used to catalog or uncatalog non-VSAM data sets on tape or dedicated disk. The data set name can have a wild card so that multiple data sets can be processed with a single command. (For example, userid.*.DOUG is a generic name where * is any 1- to 8-character simple name that would match any data set whose 3-part name begins with the userid and ends with the string “.DOUG”.)

The following example scratches three data sets using a 1-step batch job.

Example

```
//STEP1 EXEC VDSUTIL
//SYSIN DD *
DELETE userid.dataset1
DELETE userid.dataset2
DELETE userid.dataset3
SET MAXCC = 0
```

The last command in the VDSUTIL example (SET MAXCC = 0) causes VDSUTIL to generate a return code of zero even if it cannot scratch one or more of the data sets. This corresponds to the way DSSCR works: returning a zero return code regardless of whether the specified data sets are actually scratched. Had the SET command been left out, then VDSUTIL would return a nonzero return code if one or more of the data sets could not be scratched.

The following example renames two data sets using a 1-step batch job.

Example

```
//STEP1 EXEC VDSUTIL
//SYSIN DD *
ALTER userid.dataset1 NEWNAME(userid.dataset1.rename)
ALTER userid.dataset2 NEWNAME(userid.dataset2.rename)
SET MAXCC = 0
```

The following example changes the management class for multiple data sets. The first data set is placed into the DISK2YR management class; the second data set is placed into the TEMP management class.

Example

```
//STEP1 EXEC VDSUTIL
//SYSIN DD *
ALTER userid.dataset1 MANAGEMENTCLASS(DISK2YR)
ALTER userid.dataset2 MANAGEMENTCLASS(TEMP)
SET MAXCC = 0
```

The following example will catalog a data set on tape.

Example

```
//CATLG EXEC VDSUTIL
//SYSIN DD *
DEFINE NONVSAM (NAME(userid.dataset1) -
                DEVICETYPE(3490) VOLUMES(volser))
```

The following example will uncatalog the data set cataloged in the previous example.

Example

```
//UNCAT EXEC VDSUTIL
//SYSIN DD *
DELETE userid.dataset1 NOSCRATCH
```

4.3.2 CATDS

The procedure CATDS catalogs and uncatalogs data sets on tape or dedicated disks, using the IBM program IEHPROGM. Data sets on public disk volumes must be cataloged. CATDS can be used only for non-VSAM data sets on tape or dedicated disk volumes.

IBM recommends that you use the IDCAMS program instead of IEHPROGM. IDCAMS is the program executed by the VDSUTIL procedure (see section 4.3.1).

These functions can also be accomplished with the ISPF panel 3.4 or the TSO CATALOG and UNCATALOG commands. The step name for the CATDS procedure is SCRATCH. No DD statements for the data sets are required to execute this procedure. The control statement for the catalog function has the format:

```
Col. 2
CATLG DSNNAME=userid.name ,VOL=devicetype=serial
```

DSNAME and VOL must be spelled exactly as shown. The devicetype for this procedure must be a generic class designation, i.e.,

3490	standard 3490 virtual tape
3480	18-track 38000 BPI tape cartridge
3400-6	9-track 6250 or 1600 BPI tape
3390	online 3390 disk

(Esoteric names like FRGN3480 and TAPE are not permitted.)

Example

```
//stepname EXEC CATDS
//SYSIN DD *
CATLG DSNNAME=userid.name1 ,VOL=3480=serial1
CATLG DSNNAME=userid.name2 ,VOL=3400-6=(serial3,1,serial4,1)
CATLG DSNNAME=userid.name3 ,VOL=3400-6=(serial5,1)
CATLG DSNNAME=userid.name4 ,VOL=3400-6=(serial5,2)
```

The data set to be cataloged by the second CATLG statement extends over 2 volumes, so the serial numbers must be enclosed in parentheses and listed in order of creation. The third and fourth CATLG statements illustrate a catalog operation for

two data sets residing on the same tape. In this case, the data set sequence number must be given after the serial number, i.e., VOL=device=(serial,seqno).

4.3.3 UNCATDS

The UNCATDS procedure uncatalogs data sets on tape or dedicated disks using the IBM program IEHPROGM.

This procedure scratches a tape data set if the data set has EXPDT=99000 coded. Otherwise, it does not delete or scratch a data set, but instead merely removes the entry from the system catalog. If the data set is retrieved by the catalog and given a disposition of DELETE, it will also be uncataloged.

IBM recommends that you use the IDCAMS program instead of IEHPROGM. IDCAMS is the program executed by the VDSUTIL procedure (see section 4.3.1).

The control statement has the format:

```
Col.2  
UNCATLG  DSNAME=userid.name
```

DSNAME must be spelled as shown and must specify the fully qualified name of the data set. Volume information is not required when uncataloging a data set.

Example

```
//stepname EXEC UNCATDS  
//SYSIN DD *  
    UNCATLG  DSNAME=userid.name1  
    UNCATLG  DSNAME=userid.name2  
    UNCATLG  DSNAME=userid.name3  
    UNCATLG  DSNAME=userid.name4
```

4.4 Comparing Data Sets

4.4.1 COMPARE

COMPARE compares two sequential data sets on disk or tape at the logical level. It uses the IBM utility IEBCOMPR. Since the comparison operates at the logical record level, the logical record lengths must be identical. The two data sets can have different block sizes and can reside on different device types.

Two data sets are considered equal if they contain the same number of records and the corresponding records are identical. On an unequal compare, record and block number and the unequal records are printed. Ten successive unequal comparisons will

terminate the job step. The program can compare fixed-length, variable-length, or undefined-length records from blocked or unblocked data sets.

The procedure name is COMPARE, the step name is CMP. SYSUT1 and SYSUT2 are the ddnames of the input data sets. The defaults are DISP=SHR for SYSUT1 and SYSUT2.

Example

```
//stepname EXEC COMPARE
//CMP.SYSUT1 DD UNIT=TAPE,VOL=(PRIVATE,SER=tapeser),
// DSN=userid.dsname1
//CMP.SYSUT2 DD DSN=userid.dsname2
```

Two sequential data sets, one on tape and the other on a public volume, are being compared. DCB information is supplied by the labels and default dispositions are used.

4.5 Copying Data Sets

This section describes the COPY, DSCOPY, and IEBCOPY procedures.

Copying can also be done using SORT (see section 4.11.1), EDSSORT (see section 4.13.4), PDSCOPY (see section 4.15.6), DDSCOPY (see section 4.12.4), and USRPDSC (see section 4.15.7).

4.5.1 COPY

COPY copies a sequential data set on disk or tape or creates a sequential data set from a member of a partitioned data set. It uses the IBM utility ICEGENER.

COPY has the following restrictions:

- It will not work properly to copy a member of a PDS that is a load module. Use IEBCOPY (see section 4.5.3) or USRPDSC (see section 4.15.7) to copy members that are load modules.
- It cannot be used with an input tape that contains ASCII VB records. For help in copying such data, contact the NIH Help Desk.

The input and output data sets may have fixed, variable, or undefined length records, and they may have different blocking factors. Whenever LRECL, RECFM, or BLKSIZE is coded for the output data set, all three must be given.

SYSUT1 is the ddname of the input data set; SYSUT2 is the ddname of the output data set. If DCB information is not specified for the output data set, the DCB for the input data set is used and a warning message is issued. The defaults are DISP=SHR for SYSUT1 and DISP=(NEW,KEEP,DELETE) for SYSUT2.

The following example copies a data set and reblocks it..

Example

```
//stepname EXEC COPY
//COPY.SYSUT1 DD DSN=userid.dsname1,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//COPY.SYSUT2 DD DSN=userid.dsname2,
// SPACE=(TRK,(5,1),RLSE),UNIT=LONGTERM,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=11440),
// DISP=(NEW,CATLG)
```

The following example copies one member of a PDS as a sequential data set onto a scratch tape that is kept for 7 days. Only one member at a time can be copied using this method.

Example

```
//stepname EXEC COPY
//COPY.SYSUT1 DD DSN=userid.dsname1(member),
// DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//COPY.SYSUT2 DD UNIT=TAPE,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=11440),RETPD=7,
// DSN=userid.dsname2
```

In the following example the input data set, which is written in ASCII code and has a non-standard data set name, is on a foreign 9-track, 6250 BPI, ANSI-labeled tape which is handed in at Output Distribution Services. The output data set, written in EBCDIC code, is on a standard-labeled scratch tape that will be kept until the data set is uncataloged. Note that the blocking factor is changed.

Example

```
//stepname EXEC COPY
//COPY.SYSUT1 DD UNIT=FRGN6250,VOL=(PRIVATE,SER=tapeser),
// DSN='nonstandard.name',LABEL=(1,AL),EXPDT=98000,
// DCB=(OPTCD=Q,RECFM=FB,LRECL=80,BLKSIZE=800)
//COPY.SYSUT2 DD UNIT=TAPE,VOL=PRIVATE,DSN=userid.name,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=11440)
// DISP=(NEW,CATLG),EXPDT=99000
```

For further information on the ICEGENER utility, see the *DFSORT Application Programming Guide*, SC33-4035.

4.5.2 DSCOPY

DSCOPY copies a sequential data set on disk or tape. DSCOPY is an NIH-written utility.

DSCOPY has the following restrictions:

- It cannot be used to change the output values for RECFM, LRECL, or BLKSIZE
- It cannot copy multi-volume data sets.

The ddnames IN and OUT describe the input and output data sets.

Required Symbolic Parameters

NAME Names the input data set; also names the output data set unless NEWNAME is specified.

Optional Symbolic Parameters

NEWNAME Names the output data set.

INDISK Supplies the input volume serial number. Required only for a non-cataloged data set.

OUTDISK Supplies the output volume serial number. Required if the data set is to be written to a dedicated disk.

INTAPE Replaces INDISK if the input volume is a tape.

OUTTAPE Replaces OUTDISK if the output volume is a tape.

INSTOR Provides a unit name for input (such as FRGN3480) if the data set is not cataloged.

OUTSTOR Provides a unit name for output (such as FRGN3480, NOBACKUP, or DISK2YR); the default is LONGTERM.

UNITS Gives the units of allocation for the SPACE parameter; the default is TRK (tracks).

SIZE Gives the primary space allocation for a disk output data set; the default is 10.

INCR Gives the number of units in each secondary allocation; the default is 5.

UNUSED	Determines whether unused space is released; the default is RLSE. Code UNUSED= to retain unused space.
STATUS	Specifies whether the output data set is OLD or NEW; NEW is the default.
INDEN	Specifies a code (3 or 4) to indicate the density for an input 9-track tape.
OUTDEN	Specifies a code (3 or 4) to indicate the density for an output 9-track tape.
INLBL OUTLBL	Defines the type of label on a tape. SL (standard-labeled) is the default. NL (unlabeled) may be specified, but DCB information must be supplied if the input tape is NL.
LRECL	Gives the logical record length for the input data set; it is required if INLBL=NL and should not be used otherwise.
BLKSIZE	Gives the block size for the input data set; it is required if INLBL=NL and should not be used otherwise.
RECFM	Gives the record format for the input data set; it is required if INLBL=NL and should not be used otherwise.
INSEQ OUTSEQ	Specifies a sequence number for a tape data set; the default is 1.
STEPEND	Indicates the final (end of step) disposition for the output data set; CATLG is the default.

In the example below, a cataloged sequential data set will be copied to a new data set in the TEMP management class. It will be automatically scratched after 7 days. The data set is given a new name because the catalog will not permit duplicate entries.

Example

```
//stepname EXEC DSCOPY,NAME='userid.dsname1',
// NEWNAME='userid.dsname2',OUTSTOR=TEMP
```

The following example will copy a data set from a 3490 cartridge tape from another computer center to a standard virtual tape that will be kept for 30 days.. The data set will be renamed. DCB characteristics are given for the input data set.

Example

```
//stepname EXEC DSCOPY,NAME='dsname1',INTAPE=tapeser,  
// INSTOR=FRGN3490,  
// NEWNAME='userid.dsname2',RETPD=30,  
// OUTSTOR=TAPE,RECFM=FB,LRECL=80,BLKSIZE=11440  
//IN DD EXPDT=98000
```

4.5.3 IEBCOPY

IEBCOPY unloads a PDS from disk to tape or reloads a PDS from tape to disk. It can also copy all or selected members of a PDS to a new or existing PDS. IEBCOPY is an IBM utility.

IEBCOPY can also merge PDSs, select specific members of a PDS to be copied, loaded, or unloaded, replace or rename members of a PDS, copy and reblock load modules, and pretty much do anything except mash potatoes, although IBM programmer Peter Somerville says it can do that too using the little-known option /m. For information on the IEBCOPY utility, see the *DFSMS/MVS Utilities* manual.

4.6 Erasing Sensitive Data

4.6.1 ADSERASE

ADSERASE erases a disk data set or an entire tape. This procedure is quite inexpensive and should be used to eliminate the possibility of unauthorized access to data protected by the Privacy Act that is no longer needed. It works on sequential, partitioned, and direct data sets and components of VSAM data sets. ADSERASE is an NIH-written utility.

ADSERASE writes messages to the SYSPRINT data set indicating what is being erased and whether or not it was successful.

Required Symbolic Parameters

NAME Specifies the name of the disk data set or the first data set on the tape that will be erased.

Optional Symbolic Parameters

DISK Specifies the volume serial number of the disk containing the data set. Required if the data set is not cataloged.

TAPE Specifies the volume serial number of the tape containing the data set. Required if the data set is not cataloged. Omitting it will cause the first data set on the tape to be uncataloged.

STORAGE Specifies the unit name. Required if the data set is not cataloged.

STEPEND Specifies the final disposition; the default is DELETE.

LABEL Specifies the label type for tapes; the default is SL.

The following example shows how to erase a cataloged disk data set and delete the data set at the end of the step.

Example

```
//stepname EXEC ADSERASE,NAME='userid.dsname'
```

The output messages would appear as follows:

```
ADSERASE03 ERASING DISK DATA SET userid.dsname
           ON VOLUME xxxxxxx
ADSERASE05 ERASING COMPLETE
```

To erase a tape, only the first data set name must be coded. Note that the entire tape will be erased. The following example will erase the data on tape 123456 if the first data set is named userid.DATA1. (If an unlabeled tape were being erased, any data set name could be specified.)

Example

```
//stepname EXEC ADSERASE,TAPE=123456,
//           NAME='userid.DATA1',
//           STORAGE=TAPE
```

The output messages would appear as follows:

```
ADSERASE02 ERASING TAPE 123456
ADSERASE05 ERASING COMPLETE
```

To use ADSERASE on a VSAM data set, you must specifically erase each component of the data set separately (although there is probably no need to erase an index component since all it contains is compressed keys). For example, to erase the VSAM KSDS whose cluster name is userid.VSAM.MAS.BFILE, the following jobsteps must be executed:

Example

```
// EXEC ADSERASE,NAME='userid.VSAM.D.MAS.BFILE',STEPEND=
// EXEC ADSERASE,NAME='userid.VSAM.I.MAS.BFILE',STEPEND=
```

Both job steps will produce these messages:

```
ADSERASE03 ERASING DISK DATA SET userid.VSAM.D.MAS.BFILE
ON VOLUME xxxxxxx
```

```
ADSERASE03 ERASING DISK DATA SET userid.VSAM.I.MAS.BFILE
ON VOLUME xxxxxxx
```

```
ADSERASE05 ERASING COMPLETE
```

STEPEND=DELETE, which is the default in ADSERASE, is not permitted for a VSAM component and will result in an ABEND if not overridden. To delete the data set, you can code STEPEND=, as shown in the example above.

After the ADSERASE steps are executed, the VDSUTIL utility or AMS DELETE command can be used to delete (scratch) the VSAM cluster. VDSUTIL and AMS DELETE are fully described in the manual *Using VSAM and AMS at NIH*.

4.7 Printing Data Sets

This section describes the DSLIST and PRINT procedures.

Printing can also be done using EDSLIST (see section 4.13.3), PDSLIST (see section 4.15.10), DDSLIST (see section 4.12.3) and USRPDSL (see section 4.15.4).

4.7.1 DSLIST

DSLIST prints an Edit-format or non-Edit format sequential data set from disk or tape. It uses the EDSUTIL program. If the record length is greater than 132, the record will be continued on following lines until it has all been printed.

Required Symbolic Parameters

NAME Names the data set to be listed.

Optional Symbolic Parameters

DISK Specifies the volume serial number of the disk on which the data set is stored. It is not required if the data set is cataloged.

TAPE Replaces DISK if the input volume is a tape.

STORAGE Gives the unit name for the input device if it is not cataloged.

DEN Specifies a code (3 or 4) to indicate the density for a 9-track tape.

SEQ Gives the sequence number for a tape data set; the default is 1.

LBL Defines the type of label on the input tape. SL (standard-labeled) is the default.

LRECL	Gives the logical record length for the input data set; it is required if LBL=NL and should not be used otherwise.
BLKSIZE	Gives the block size for the input data set; it is required if LBL=NL and should not be used otherwise.
RECFM	Gives the record format for the input data set; it is required if LBL=NL and should not be used otherwise.
OPTIONS	The options, which may be specified for the output listing described in section 1 and in the <i>Edit Format Utility Package</i> manual. For example, CC uses the first character of each record for ANSI carriage control, INDENT=10 inserts 10 blanks at the beginning of each line, SKIP 20 skips the first 20 lines, and TAKE 50 prints 50 lines.
UTIOPT	Specifies parameters passed to the EDSUTIL program to affect message and ABEND processing. For further information, consult the <i>Edit Format Utility Package</i> manual.

To list a cataloged sequential data set from a disk volume, use the following job control language:

Example

```
//stepname EXEC DSLIST,NAME='userid.dsname'
```

The following example will list the second data set from a standard virtual tape.

Example

```
//stepname EXEC DSLIST,NAME='userid.dsname',
//          TAPE=tapeser,STORAGE=TAPE,SEQ=2
```

4.7.2 PRINT

PRINT prints a sequential data set from disk or tape. It can select and format the input, print in character or hexadecimal format, and convert from packed decimal. It uses the IBM utility IEBTPCH.

Under the standard format, each new logical record begins on a new line. The end of a logical record is marked by an asterisk and the end of a block by two asterisks. Each line contains up to 96 characters, divided into 12 groups of 8 characters each and separated by 2 blanks. Therefore, if the input to the basic print operation is:

```
THIS IS WHAT THE DEFAULT OUTPUT LOOKS LIKE
```

the output will be:

```
THIS IS WHAT THE DEFAULT OUTPUT LOOKS LI KE*
```

Unprintable characters appear as blanks. Because of this default format, users may find it more convenient to use one of the other print utilities.

The default DCB attributes of the SYSUT2 output data set are:
RECFM=FBA,LRECL=121,BLKSIZE=121

Note: the sum of the field lengths in any IEBTPCH record statement may not exceed the output logical record length minus 1. Also, the maximum line size is 144.

The DD statement SYSUT1 defines the input data set. The default is a sequential data set with no editing of the output data set. The stepname is PRINT.

For more information on the IEBTPCH utility control statements, see the *IBM DFSMS/MVS Utilities* manual, SC26-4926.

Example

```
//stepname EXEC PRINT
//PRINT.SYSUT1 DD UNIT=FRGN3490,DISP=(OLD,KEEP),
//      DSN=userid.dsname,VOL=(PRIVATE,SER=tapeser),
//      LABEL=(1,NL),EXPDT=98000
//      DCB=(RECFM=FB,LRECL=80,BLKSIZE=11440)
```

In this example the first data set on a foreign 3490 cartridge tape will be dumped to the printer. Since there is no label, DCB information must be supplied. If a standard-labeled tape is used, the DCB information is not needed on the DD statement. The data will be printed in standard format.

Frequently users want to print a tape, editing the data to eliminate the blanks between each group of eight characters. The following example illustrates how to do this and start printing after the 250th record. In this example the FIELD parameter takes advantage of all the defaults for starting input and output locations. It will read the first 80 bytes of the input record and start writing in the first column on the print line.

Example

```
//stepname EXEC PRINT
//PRINT.SYSUT1 DD DSN=userid.dsname,UNIT=TAPE,
//      DISP=(OLD,KEEP),VOL=(PRIVATE,SER=tapeser)
//PRINT.SYSIN DD *,DSN=
//      PRINT STRTAFT=250,MAXFLDS=1
//      RECORD FIELD=(80)
```

No editing is allowed when printing variable spanned (VS) or variable blocked spanned (VBS) records. These data sets must be printed with the 8 character, 2 blank format.

If the LRECL of the data set is greater than 144, editing can still be done if the user chooses as the LRECL a valid line size (no more than 144) which can be evenly divided into the block size. In the case of a data set with an LRECL of 260 and a block size of 5200, each logical record can be broken into 2 print lines of 130 characters each. To accomplish this, DCB parameters must be specified for the input (SYSUT1) data set and IEBPTPCH control statements must be provided. DCB parameters must also be provided for the output (SYSUT2) data set since they exceed the default (121 output record characters per line). Notice that the LRECL is increased by 1 to allow for the carriage control. The input data set is cataloged.

Example

```
//stepname EXEC PRINT
//PRINT.SYSUT1 DD DSN=userid.dsname,DISP=SHR,
//          DCB=(RECFM=FB,LRECL=130,BLKSIZE=5200)
//PRINT.SYSUT2 DD DCB=(RECFM=FBA,LRECL=131,BLKSIZE=131)
//PRINT.SYSIN DD *,DSN=
PRINT MAXFLDS=1
RECORD FIELD=(130)
```

This method may be used for data sets with either fixed or fixed blocked records. If the data set has fixed-length unblocked records, RECFM=FB must be specified on the SYSUT1 DD statement.

The following JCL will cause a tape data set to be printed in hexadecimal:

Example

```
//stepname EXEC PRINT
//PRINT.SYSUT1 DD UNIT=TAPE,DISP=(OLD,KEEP),
//          VOL=(PRIVATE,SER=tapeser),DSN=userid.dsname
//PRINT.SYSIN DD *,DSN=
PRINT TOTCONV=XE
```

The following example illustrates how to print specific members of a cataloged PDS:

Example

```
//stepname EXEC PRINT
//PRINT.SYSUT1 DD DISP=(OLD,KEEP),
//          DSN=userid.dsname
//PRINT.SYSIN DD *,DSN=
```

```
PRINT  TYPORG=PO , MAXNAME=3 , MAXFLDS=2
MEMBER NAME=DATA1
RECORD FIELD=( 100 , 1 , , 1 )
MEMBER NAME=DATA2
MEMBER NAME=SORCE
RECORD FIELD=( 80 , 10 , , 20 )
```

Three members of a partitioned data set are to be printed. DCB information will be retrieved from the volume table of contents on the disk. (The logical record length happens to be 100). The control statements identify the three members to be printed and the editing to be performed. The first member, DATA1, will be printed with one logical record, 100 bytes per line beginning with column 1 of the input record, starting in column 1 of the print line. The second member, DATA2, will be printed with no format control. The third member, SORCE, will have only 80 bytes of each record printed; the rest of the logical record is to be ignored. Processing begins with the tenth byte of the input record; the printed output begins in column 20 of the print line.

4.7.3 WYLABEL

The WYLABEL procedure prints labels from a single column of data. It is an NIH-written program. The labels can be either

- three-across adhesive labels with a maximum of 8 lines per label (4" wide x 1 7/16" deep and 38 characters per line)
- or four-across Cheshire labels with a maximum of five lines per label. (3 1/4" wide x 1" deep and 31 characters per line)

A procedure is provided which allows the user to specify the form type, channel control, and whether the program or the user will supply the alignment guide.

Input Data Set

The attributes of the input data set must be RECFM=FB, LRECL=44, and BLKSIZE=1760. Place the labels in one column and separate them with a single blank line.

Required Symbolic Parameters

IPDSN Names the input data set.

Optional Symbolic Parameters

FORM Identifies the label on which the data set will be printed. Use 0609 for adhesive labels or 0593 for Cheshire labels. The default is 0609.

CHAN Defines the Forms Control Buffer (FCB) used to set the vertical

spacing. For adhesive labels, use 6609 for 6 LPI or 8609 for 8 LPI. For Cheshire labels you must use 6593 for the labels to be properly aligned. The default is 6609.

ALIGN Specifies whether you want a standard set of Xs to be printed before your data set and used by the operators as an alignment guide or whether your data set starts with an alignment guide. YES used the standard guide and NO uses the guide in your data set.

COPY The number of copies of the data set to be printed. The default is one.

The following example prints the data set on adhesive labels.

```
//stepname EXEC WYLABEL,IPDSN='userid.dsname'
```

The next example prints the data set on Cheshire labels.

```
//stepname EXEC WYLABEL,FORM=0593,CHAN=6593,  
// IPDSN='userid.dsname'
```

4.8 Recalling Data Sets via Batch

4.8.1 IEFBR14

IEFBR14 performs no action other than give the return code 0, but the job scheduler checks JCL statements for syntax errors, allocates space for data sets and performs disposition processing.

The following example recalls a large number of data sets that have been migrated (staged to level 1 or 2). No more than 300 data sets should be recalled in a single batch job. The larger the data sets, the fewer the number that can be recalled in the batch job.

Example

```
//stepname EXEC PGM=IEFBR14  
//DD1 DD DSN=userid.dsn1,DISP=SHR  
//DD2 DD DSN=userid.dsn2,DISP=SHR  
//DD3 DD DSN=userid.dsn3,DISP=SHR  
.  
.  
.  
//DDn DD DSN=userid.dsn300,DISP=SHR
```

4.9 Renaming Disk Data Sets

4.9.1 DSRENAME

DSRENAME renames a sequential data set or PDS on disk. It is an NIH-written utility. If a data set is referenced through the catalog, the old data set name will be uncataloged and the new data set name will be cataloged. This procedure may be used to rename sequential, partitioned, and direct data sets. It cannot be used for VSAM data sets. See the publication *Using VSAM and AMS at NIH* for further information. For online disk data sets, the TSO RENAME command or ISPF panel 3.4 may also be used to perform this function.

DSRENAME cannot receive a data set passed from a previous step.

A step completion code of 000 indicates that the rename operation has completed successfully. Error conditions are indicated by a step completion code of 008 and an explanatory message in the printed output of the DSRENAME step.

Required Symbolic Parameters

NAME Gives the original name of the data set.

NEWNAME Gives the new data set name.

Optional Symbolic Parameters

DISK Supplies the volume serial number of the disk. Required if the data set is not cataloged.

STORAGE Provides a unit name. Required if the data set is not cataloged.

The following example renames and recatalogs a data set.

Example

```
//stepname EXEC DSRENAME,NAME='aaaaiii.oldname',  
// NEWNAME='aaaaiii.newname'
```

4.10 Scratching Disk Data Sets

This section describes the DSSCR and DSSCRV procedures.

4.10.1 DSSCR

DSSCR scratches a single sequential data set or PDS on disk. If the data set is cataloged, it will be uncataloged. Even if the data set does not exist, it will be uncataloged. DSSCR can also uncatalog (but not delete) a data set cataloged on a 3490 or 3480 tape. DSSCR is an NIH-written utility.

DSSCR cannot receive a data set passed from a previous step.

Required Symbolic Parameters

NAME Names the data set to be scratched.

Optional Symbolic Parameters

DISK Supplies the volume serial number of a dedicated disk. Required unless the data set is cataloged.

STORAGE Gives the unit name.

The following example scratches a cataloged data set.

Example

```
//stepname EXEC DSSCR,NAME='userid.dsname'
```

4.10.2 DSSCRV

DSSCRV scratches a sequential data set or PDS on a specific disk volume without using the catalog to determine the location of the data set. DSSCRV honors the volume request and scratches the dataset specified in the NAME parameter on the volume specified in the DISK parameter. If the data is cataloged on that volume, it also will be uncataloged.

Required Symbolic Parameters

NAME Names the data set to be scratched.

DISK Supplies the volume serial number of the disk.

The following example scratches a non-cataloged data set from DIST01.

Example

```
//stepname EXEC DSSCRV,NAME='userid.dsname',  
// DISK=DIST01
```

4.11 Sorting and Merging Data Sets

This section describes the SORT and MERGE procedures.

Sorting can also be done using EDSSORT (see section 4.13.4).

For more information on SORT and MERGE, see *DFSORT Application Programming Guide*, SC33-4035.

4.11.1 SORT

SORT sorts and copies sequential data sets on disk or tape. It uses the IBM utility DFSORT. The CHECKPOINT/RESTART option is not supported.

Before attempting to override parameters in any of these procedures or trying to exceed any of the limitations they impose, consult the *DFSORT Application Programming Guide*, SC33-4035. When calling the DFSORT utility, PGM= SORT must be used for any stand-alone sort not invoking a NIH Computer Center procedure. The following statement must be included for the sort messages if the DFSORT program is called from a language unless a NIH Computer Center procedure is being used:

Example

```
//stepname.SORTMSGs DD SYSOUT=A
```

The DFSORT program can process VSAM and sequential data sets. Blocked or unblocked, fixed or variable length records can be sorted.

The NIH Computer Center does not provide procedures for using tape sort work areas (intermediate storage) because using disk is cheaper and reduces the possibility of I/O errors. When necessary, very large jobs should be divided into smaller jobs so they can be accommodated within the scratch disk limit.

The SORTIN DD statement describes the input data sets; the SORTOUT DD statement describes the output data set.

Non-VSAM data sets on unlike devices may be concatenated in the SORTIN DD statement. All data sets in the concatenation must have the same record length and record format. The block size may vary, but the data set with the largest block size must be specified on the first DD statement in the concatenation.

The SORT control statement begins in any column after column 1 and has the format given below. The second example of syntax would be used if all fields to be sorted were of the same format.

```
SORT FIELDS=(col , len , format , order , col , len , format , order . . . ) ,  
          FILSZ=numrec
```

or

```
SORT FIELDS=(col , len , order , col , len , order . . . ) , FORMAT=format ,  
          FILSZ=numrec
```

where

col the beginning column of the control field; for variable length records, add 4 for the 4-byte record descriptor word (RDW) at the beginning of each record

len the length of the control field.

format the format of the data in the control field.

CH	character
ZD	zoned decimal
PD	packed decimal
FI	fixed point
BI	binary
FL	floating point

HINT: To sort on a numeric field that might contain blanks, specify the character (CH) format.

order A for ascending order
 D for descending order

numrec a count of the number of records to be sorted. The sort will terminate if this count does not agree with the number of records read. If the exact number is not known, code FILSZ=Enumrec, *E* indicating estimate.

The parameters *col*, *len*, and *order* are given once for each field to be used to sort the data set. If the SORT statement is to be continued onto the next statement, it should be broken after a comma.

SORT sorts up to approximately 1,650,300 80-byte records and uses 3 sort work areas on 3 scratch disks. The default space allocation for the SORTWRK areas is 6 cylinders primary and 7 cylinders secondary, i.e., SPACE=(CYL,(6,7),RLSE). The SORTWK01, SORTWK02, and SORTWK03 DD statements have a default value of UNIT=SYSDA. The step name is SORT.

Using recommended block sizes (see the BLKSIZE subparameter in section 2.6.4) is particularly important for data sets that will be sorted. SORT reads and writes data with EXCP processing rather than the standard access methods. While this technique permits great sorting efficiency for reasonable block sizes, it is quite inefficient when sorting data with small block sizes (e.g., 400).

Example

```
//stepname EXEC SORT  
//SORT.SORTIN DD DSN=&&dsname,DISP=(OLD,DELETE),
```

```

//      DCB=(RECFM=FB,LRECL=120,BLKSIZE=11400)
//SORT.SORTOUT DD UNIT=DISK2YR,DISP=(,CATLG),DCB=*.SORTIN,
//      SPACE=(TRK,(20,1),RLSE),DSN=userid.dsname
//SORT.SYSIN DD *
      SORT FIELDS=(10,7,CH,A),FILSZ=1254

```

In this example SORTIN is a temporary data set passed from a previous step. It will be deleted at the end of this step. The records are fixed length, 120 characters long, 95 records to a block. The SORTOUT data set is to be placed in the DISK2YR management class and cataloged with a primary allocation request of 20 tracks. DCB information is copied from the SORTIN DD statement. The records will be sorted in ascending order on a seven-byte field beginning in column 10. There are exactly 1254 records in the input data set. The sort will terminate if the number of records read in does not agree with this value. If this value were an approximation of the number of records in the data set, FILSZ=E1254 could be used.

4.11.2 MERGE

MERGE merges sequential data sets on disk or tape. It uses the IBM utility DFSORT. MERGE accepts up to 100 data sets as input. These data sets must already be ordered on the fields described in the MERGE control statement. The input ddnames must be SORTIN00 through SORTIN99, and they must be in ascending sequence. SORTOUT is the ddname for the output data set. The records may be blocked or unblocked, fixed or variable length, but for a given application the record formats must be the same. If unlike block sizes are to be merged, the largest block size should be first. The stepname is MERGE.

The MERGE control statement begins in any column after column 1 and has the format given below. The second example of syntax would be used if all fields to be sorted were of the same format.

```

MERGE FIELDS=(col,len,format,order,col,len,format,order...),
        FILSZ=numrec

```

or

```

MERGE FIELDS=(col,len,order,col,len,order...),FORMAT=format,
        FILSZ=numrec

```

where

col the beginning column of the control field; for variable length records, add 4 for the 4-byte record descriptor word (RDW) at the beginning of each record

len the length of the control field.

format the format of the data in the control field.

CH	character
ZD	zoned decimal
PD	packed decimal
FI	fixed point
BI	binary
FL	floating point

order A for ascending order
 D for descending order

numrec a count of the number of records in all the input data sets which are to be merged. The merge will terminate if this count does not agree with the number of records read. If the exact number is not known, code FILSZ=Enumrec, *E* indicating estimate.

Example

```
//stepname    EXEC   MERGE
//MERGE.SORTIN01 DD DSN=userid.dsname,UNIT=TAPE,
//       VOL=(PRIVATE,SER=tapeser),DISP=SHR,
//       DCB=(RECFM=FB,LRECL=133,BLKSIZE=11438,DEN=4)
//MERGE.SORTIN02 DD DSN=&&dsname,DISP=(OLD,DELETE),
//       DCB=*.SORTIN01
//MERGE.SORTOUT DD DSN=userid.dsname,UNIT=TAPE,
//       DISP=(NEW,KEEP,KEEP),VOL=PRIVATE,
//       DCB=*.SORTIN01,RETPD=21
//MERGE.SYSIN  DD  *
          MERGE FIELDS=(20,10,CH,D),FILSZ=E120000
          END
```

In this example two data sets are to be merged. One is from a tape, and the other is a temporary disk data set that has been passed from a previous step and will be deleted at the end of this step. The merged output is placed on a scratch tape which will be kept for 21 days. The input data sets will be merged in descending order on the ten-byte control field beginning in column 20 of each record. The total number of records from both input data sets is approximately 120,000.

4.12 Direct Access Data Sets

4.12.1 DDSSAVE

DDSSAVE unloads a direct access data set from disk to tape.

Internally, the DDSSAVE procedure invokes the NIH-written program, DSCOPYJ, which in turn links to the IBM utility program, IEHMOVE.

Required Symbolic Parameters

NAME Names the input data set; also names the output data set unless NEWNAME is specified.

Optional Symbolic Parameters

DISK Supplies the input volume serial number. Required if the data set is not cataloged.

TAPE Supplies the output volume serial number. The default is to use a scratch tape.

NEWNAME Names the output data set.

STORAGE Gives a unit name for input. Required if the PDS is not cataloged.

STATUS Specifies whether the output data set is OLD or NEW; NEW is the default.

DRIVE Gives a unit name for output; the default is TAPE.

DEN Specifies a code (3 or 4) to indicate the density for a 9-track tape.

RETPD Specifies the number of days to retain the data set on tape. Required unless overwriting an existing data set.

SEQ Gives the sequence number for the tape data set; the default is 1. Overwriting data set 1 destroys any subsequent data sets.

LBL Defines the type of label on the output tape. SL (standard-labeled) is the default. NL (unlabeled) may be specified for foreign tapes.

STEPEND Indicates the final (end of step) disposition for the output data set; KEEP is the default.

This example saves a cataloged direct data set from a disk volume to a virtual scratch tape. The data set is to be kept for 6 days.

Example

```
//stepname EXEC DDSSAVE,  
// NAME='userid.dsname',RETPD=6
```

This example saves a cataloged direct data set from a disk volume as the second data set on a virtual tape. The data set is to be kept for 8 days.

Example

```
//stepname EXEC DDSSAVE,NAME='userid.dsname',  
// TAPE=tapeser,SEQ=2,RETPD=8
```

This example saves a direct data set to a foreign 3490 cartridge tape to be sent to another installation.

Example

```
//stepname EXEC DDSSAVE,NAME='userid.dsname',  
// DRIVE=FRGN3490,  
// TAPE=tapeser  
//OUT DD EXPDT=98000
```

4.12.2 DDSGET

DDSGET reloads a direct access data set from tape to disk. Internally, the DDSGET procedure invokes the NIH-written program DSCOPYJ, which in turn links to the IBM utility program IEHMOVE.

The IBM utility program IEHMOVE is old and out of date. IBM provides it for the sake of compatibility only and recommends that you use IEBCOPY or DFSMSdss instead.

This procedure has a problem when the output data set is going to a public disk. Unless you override the space request, the step will get a condition code of 8 and you will see the following message in your output "IEH478I MOVE/COPY REQUEST IGNORED BECAUSE TARGET VOLUME(S) MANAGED BY THE STORAGE MANAGEMENT SUBSYSTEM." The cause of the problem is the inability of the IBM utility program IEHMOVE to dynamically allocate new data sets on system managed volumes. To circumvent this problem, override any one of the space parameters (i.e., UNITS= or SIZE=) for the output data set to change it from its default value.

Required Symbolic Parameters

NAME Names the input data set; also names the output data set unless
NEWNAME is given.

Optional Symbolic Parameters

NEWNAME Names the output data set.

TAPE	Supplies the input volume serial number. Required if the input data set is not cataloged.
STORAGE	Gives a unit name for output; the default is LONGTERM.
STATUS	Specifies whether the output data set is OLD or NEW; NEW is the default.
DRIVE	Gives a unit name for input; the default is TAPE.
DEN	Specifies a code (3 or 4) to indicate the density for a 9-track tape.
SEQ	Gives the sequence number for the tape data set; the default is 1.
LBL	Defines the type of label on the tape. SL (standard-labeled) is the default. NL (unlabeled) may be specified.
STEPEND	Indicates the final (end of step) disposition for the output data set; CATLG is the default.
DISK	Supplies the output volume serial number. Required if the data set is to be written to a dedicated disk.

The following parameters are used with a non-specific volume request to select a volume with enough available space to store the data set. Most data sets can be accommodated by the default values. For a very large data set, they may need to be increased.

UNITS	Gives the units of allocation for the SPACE parameter; the default is TRK (tracks).
SIZE	Specifies the number of units in the primary allocation; the default is 10.

This example retrieves a direct data set from a virtual tape to a data set in the LONGTERM management class. the new data set will be cataloged and given the same name as the tape data set.

Example

```
//stepname EXEC DDSGET,NAME='userid.dsname',
// TAPE=tapeser,SIZE=11
```

This example retrieves a direct data set from the third data set on a virtual tape to a data set in the TEMP management class.

Example

```
//stepname EXEC DDSGET,NAME='userid.dsname',  
// STORAGE=TEMP,UNITS=CYL,TAPE=tapeser,SEQ=3
```

4.12.3 DDSLIST

DDSLIST lists a direct access data set. If the logical record length is greater than 132, the record will be continued on following lines until it has all been printed. The DDSLIST procedure invokes the NIH-written program DSCOPYJ.

Required Symbolic Parameters

NAME Names the data set to be listed.

Optional Symbolic Parameters

DISK Specifies the volume serial number of the disk on which the data set is stored. Required if the data set is not cataloged.

STORAGE Gives the unit name for the input device. Required if the data set is not cataloged.

OPTIONS Specifying SUPPRESS suppresses the listing of empty slots (records beginning with hexadecimal FF).

The following example will list a cataloged direct data set.

Example

```
//stepname EXEC DDSLIST,NAME='userid.dsname'
```

4.12.4 DDSCOPY

DDSCOPY copies a direct access data set from one direct access device to another. Internally, the DDSCOPY procedure invokes the NIH-written program DSCOPYJ, which in turn links to the IBM utility program, IEHMOVE.

The IBM utility program IEHMOVE is old and out of date. IBM provides it for the sake of compatibility only and recommends that you use IEBCOPY or DFSMSdss instead.

This procedure has a problem when the output data set is going to a public disk. Unless you override the space request, the step will get a condition code of 8 and you will see the following message in your output "IEH478I MOVE/COPY REQUEST IGNORED BECAUSE TARGET VOLUME(S) MANAGED BY THE STORAGE MANAGEMENT SUBSYSTEM." The cause of the problem is the inability of the IBM utility program IEHMOVE to dynamically allocate new data sets on system managed volumes. To circumvent this problem, override any one of the space parameters (i.e., UNITS= or SIZE=) for the output data set to change it from its default value.

Required Symbolic Parameters

NAME Names the input data set; also names the output data set unless
NEWNAME is specified.

Optional Symbolic Parameters

NEWNAME Names the output data set.

INDISK Supplies the input volume serial number. Required if the data set is
not cataloged.

INSTOR Provides a unit name for input. Required if the input data set is not
cataloged.

OUTSTOR Provides a unit name for output; the default is LONGTERM.

STATUS Specifies whether the output data set is OLD or NEW; NEW is the
default.

STEPEND Indicates the final (end of step) disposition for the output data set;
CATLG is the default.

OUTDISK Supplies the output volume serial number. Required if the data set is
to be written to a dedicated disk.

The following parameters are used with a non-specific volume request to select a volume with enough available space to store the data set. Most data sets can be accommodated by the default values. For a very large data set, they may need to be increased.

UNITS	Gives the units of allocation for the SPACE parameter; the default is TRK (tracks).
SIZE	Specifies the number of units in the primary allocation; the default is 10.

The following example will copy an existing direct access data set to a new data set in the LONGTERM management class and rename and catalog it.

Example

```
//stepname EXEC DDSCOPY,NAME='userid.dsname1',
// NEWNAME='userid.dsname2',SIZE=8
```

4.13 Edit Format Data Sets

Edit format is the standard format used by WYLBUR to store data sets. Each edit format line has a number which is stored adjacent to the text of the line. edit format is typically two or three times as efficient in the use of disk space as other formats.

This section describes the EDSIN, EDSOUT, EDSLIST, EDSSORT, EDSSAVE, and EDSSGET procedures.

These procedures convert to and from edit format. They each invoke the NIH-written EDSUTIL program for the conversion. The EDSUTIL program also permits manipulation of data sets without conversion to or from edit format. It is fully documented in the *Edit Format Utility Package* manual.

NOTE: Edit format data sets are not compatible with SILK (Secure Internet LinKed) technology that allows users to display OS/390 data sets through the World Wide Web. For information on SILK, visit:

<http://titan.nih.gov>

Programmers should use the edit format procedures and not try to write their own to convert to and from edit format. CIT will not pre-announce internal changes to the edit format software, and assumes no responsibility if changes to this software cause a job that invokes it illegally to fail. The internal format of WYLBUR's edit format data sets is subject to change without advance notice.

4.13.1 EDSIN

EDSIN converts an edit format data set to a non-edit format temporary data set that can be read by a batch job. The temporary data set is fixed-blocked format and named &INPUT. It is passed to a later job step.

Required Symbolic Parameters

NAME Specifies the name of the edit format data set that will be input to the program.

Optional Symbolic Parameters

DISK Specifies the volume serial number of the disk where the edit format data set is stored. Required if the data set is not cataloged.

LRECL Specifies the logical record length for the temporary data set. Cannot exceed 1000. LRECL=80 is the default.

RECFM Specifies the record format for the temporary data set. RECFM=FB is the default.

BLKSIZE Specifies the block size for the temporary data set. It must be a multiple of the LRECL. BLKSIZE=11440 is the default.

STORAGE Gives a unit name for input. Required if the data set is not cataloged.

INPUT Specifies a name for the temporary input data set. INPUT='&INPUT' is the default.

OPTIONS Many of the options which may be specified for a temporary data set are documented in section 1 and are fully described in the *Edit Format Utility Package* manual. Two of the options are LIST, to create a listing of the input data set and NUMBERED, to place the line numbers in columns 73-80.

UTILOPT Specifies parameters passed to the EDSUTIL program to affect message and ABEND processing. For further information, consult the *Edit Format Utility Package* manual.

SIZE Specifies a primary space allocation for output. The default is 5 cylinders.

For example, the following job control language statements can be used to compile a program that is stored as a cataloged WYLBUR data set:

Example

```
//stepname EXEC EDSIN,NAME='userid.dsname'  
//stepname EXEC FORTCOMP  
//COMP.SYSIN DD DSN=&INPUT,DISP=(OLD,DELETE)
```

The following job control language permits a user's program to read data from a cataloged edit format data set:

Example

```
//stepname EXEC EDSIN,NAME='userid.dsname'  
//stepname EXEC FORTCLG  
//GO.SYSIN DD DSN=&INPUT,DISP=(OLD,DELETE)
```

EDSIN normally changes the data set to card images. To change it to records of some other length, use:

Example

```
//stepname EXEC EDSIN,NAME='userid.dsname',  
// LRECL=record,BLKSIZE=block
```

When converting an edit format data set to card images, the line numbers can be placed in columns 73 through 80:

Example

```
//stepname EXEC EDSIN,NAME='userid.dsname',  
// OPTIONS=NUMBERED
```

A temporary name other than &INPUT can be used.

Example

```
//stepname EXEC EDSIN,NAME='userid.dsname',  
// INPUT='&temp'
```

where

&temp is the temporary name to be used. Remember that it must be 8 or fewer characters and must start with an ampersand.

This is useful if a job step is to read several edit format data sets since each data set must have a different dsname.

4.13.2 EDSOUT

EDSOUT converts output from a batch job to edit format. It assumes that the output is in a temporary data set named &OUTPUT that is passed from an earlier job step. The logical record length of a data set which is input to EDSOUT should not exceed 1000 (1004 for Variable format) and the block size should not exceed 32,760.

Required Symbolic Parameters

NAME Specifies the name of the output edit format data set which will be stored on disk.

Optional Symbolic Parameters

DISK Specifies the volume serial number of the disk where the edit format data set is to be stored. Required if the data set is not cataloged.

STORAGE Specifies the unit name for output; the default is LONGTERM.

STATUS OLD indicates that a data set with the same name is to be replaced. NEW is the default.

UNITS Gives the units of allocation for the SPACE parameter; the default is TRK (tracks).

SIZE Gives the primary allocation for the SPACE parameter; the default is 10.

INCR Gives the number of units in each secondary allocation; the default is 5.

UNUSED Determines whether unused space is released; the default is RLSE. Code UNUSED= to retain unused space.

OUTPUT Specifies a name for the temporary output data set. OUTPUT='&OUTPUT' is the default.

DISP PASS specifies that the temporary data set is to be used in subsequent steps. DELETE is the default.

OPTIONS Many of the options which may be specified for the input data set are documented in section 1 and are fully described in the *Edit Format Utility Package* manual. Two of the options are LIST, to create a listing of the input data set and NUMBERED, to obtain the line numbers from columns 73-80 of the input data set.

STEPEND Indicates the final (end of step) disposition for the output data set. CATLG is the default.

UTILOPT Specifies parameters passed to the EDSUTIL program to affect message and ABEND processing. For further information, consult the *Edit Format Utility Package* manual.

For example, to convert normal printer output from a FORTRAN program and save it as a cataloged edit format data set in the LONGTERM management class, use the following job control language:

Example

```
//stepname EXEC FORTCLG
//FORT.SYSIN DD *
(FORTRAN program)
/*
//GO.FT10F001 DD DSN=&OUTPUT,UNIT=SYSDA,
// SPACE=(TRK,(30,10),RLSE),DISP=(,PASS),
// DCB=(RECFM=FB,LRECL=133,BLKSIZE=11438)
//*
//stepname EXEC EDSOUT,NAME='userid.dsname'
```

A data set that already exists can be overwritten:

Example

```
//stepname EXEC EDSOUT,NAME='userid.dsname',
// STATUS=OLD
```

A different temporary data set name can be specified:

Example

```
//stepname EXEC EDSOUT,NAME='userid.dsname',
// OUTPUT='&temp'
```

where

&temp is the temporary name to be used. Remember that it must be 8 or fewer characters and must start with an ampersand.

This is useful if a job step has several output data sets that are to be converted to edit format since each one must have a different dsname.

The default step-end disposition is DELETE for the input data set. If the data set is to be used in subsequent steps, DISP=PASS should be coded.

Example

```
// EXEC EDSOUT,NAME='userid.dsname',DISP=PASS
```

Storing GO Step Output into a WYLBUR Data Set (EDSOUT)

The following general example of job control language statements can be used to store GO step output into a cataloged WYLBUR data set:

Example

```
//stepname EXEC FORTCLG
//GO.ddname DD UNIT=SYSDA,DISP=(NEW,PASS),
// DCB=(RECFM=format,LRECL=record,BLKSIZE=0),
// SPACE=(TRK,(10,5),RLSE),DSN=&OUTPUT
//GO.SYSIN DD *
      (data)
//stepname EXEC EDSOUT,NAME='userid.dsname'
```

where

- | | |
|--------|---|
| ddname | is the name defined by the program or by the language for the output data set. |
| format | is the format of the created data set, including the kind of carriage control which is either defined in the program or defined by the language. |
| record | is the number of characters per line of output plus 1 for a carriage control character, and can be changed by the programmer; it must be less than or equal to 500. |

4.13.3 EDSLIST

EDSLIST prints an edit format or non-edit format data set on disk or tape. If the logical record length is greater than 132, the record will be continued on following lines until it has all been printed. EDSLIST can print without line numbers, can print selected lines of the data set, and can use carriage control.

Required Symbolic Parameters

NAME Names the data set to be listed.

Optional Symbolic Parameters

- | | |
|---------|--|
| DISK | Specifies the volume serial number of the disk on which the data set is stored. It is not required if the data set is cataloged. |
| TAPE | Replaces DISK if the input volume is a tape. |
| STORAGE | Gives the unit name for the input device. Required if the data set is not cataloged. |
| DEN | Specifies a code (3 or 4) to indicate the density for 9-track tape. |
| SEQ | Gives the sequence number for a tape data set; the default is 1. |

LBL	Defines the type of label on the output tape. SL (standard-labeled) is the default. NL (non-labeled) may be specified.
LRECL	Gives the logical record length for the input data set; it is required if LBL=NL and should not be used otherwise.
BLKSIZE	Gives the block size for the input data set; it is required if LBL=NL and should not be used otherwise.
RECFM	Gives the record format for the input data set; it is required if LBL=NL and should not be used otherwise.
OPTIONS	Many of the options, which may be specified for the output listing are documented in section 1 and are described in the <i>Edit Format Utility Package</i> manual. For example, two of the options are CC, to use the first character of each record for ANSI carriage control and INDENT=10 to insert 10 blanks at the beginning of each line.
UTIOPT	Specifies parameters passed to the EDSUTIL program to affect message and ABEND processing. For further information, consult the <i>Edit Format Utility Package</i> manual.

To list a cataloged edit format data set, use the following job control language:

Example

```
//stepname EXEC EDSLIST,NAME='userid.dsname'
```

The following example will list a cataloged edit format data set, which has ISO/ANSI carriage control characters in column 1. Ten blanks will be inserted at the beginning of each line.

Example

```
//stepname EXEC EDSLIST,NAME='userid.dsname',
//          OPTIONS='CC,INDENT=10'
```

4.13.4 EDSSORT

EDSSORT sorts an edit format input data set and writes it to an edit format output data set. It can be used to sort alphabetical and numerical data. EDSSORT executes the EDSUTIL program, which calls the IBM DFSORT program. In a single step it converts the edit format data set to a temporary non-edit format data set, calls the DFSORT program to sort the data, and then converts the data set back to edit format.

Required Symbolic Parameters

NAME Names the input data set.

NEWNAME Names the sorted output data set.

Optional Symbolic Parameters

DISK Supplies the input volume serial number. Not required if the data set is cataloged.

NEWDISK Supplies the output volume serial number. Required only for dedicated disk files.

STORAGE Gives the unit name for the input data set. Required if the data set is not cataloged.

NEWSTOR Gives the unit name for the output data set; the default is LONGTERM.

UNITS Gives the units of allocation for the SPACE parameter; the default is TRK (tracks).

SIZE Gives the primary allocation for the SPACE parameter; the default is 10.

INCR Gives the number of units in each secondary allocation; the default is 5.

LRECL Gives the length of the longest line for the input (and output) data sets. The default is 133; longer records will be truncated unless LRECL is specified.

BLKSIZE Specifies a block size: the default is 11438. The value given must be an even multiple of the LRECL.

UNUSED Determines whether unused space is released; the default is RLSE. Code UNUSED= to retain unused space.

STATUS OLD indicates that the output data set already exists and should be written over; the default is NEW.

UTILOPT Specifies parameters passed to the EDSUTIL program to affect message and ABEND processing. For further information, consult the *Edit Format Utility Package* manual.

STEPEND Indicates the final (end of step) disposition for the output data set; CATLG is the default.

Example

```
//stepname EXEC EDSSORT,NAME='userid.dsname',  
//          NEWNAME='userid.newdsname'  
//EDSSORT.SYSIN DD *  
SORT  FIELDS=(col,len,order,...),FORMAT=CH  
/*
```

where

col the starting column number of the sort field

len the length of the sort field.

order A for ascending order
D for descending order.

The parameters *col*, *len*, and *order* are given once for each sort field. The SORT statement must begin after column 1. If it is to be continued onto the next statement, it should be broken after a comma and column 72 must contain a non-blank character. The continuation statement must start in column 16.

EDSSORT assumes that the new data set does not already exist. To write over an existing data set, use STATUS=OLD, as shown below.

Example

```
//stepname EXEC EDSSORT,NAME='userid.dsname',  
//          NEWNAME='userid.newdsname',STATUS=OLD  
//EDSSORT.SYSIN DD *  
SORT  FIELDS=(15,10,A),FORMAT=CH
```

The default amount of space allocated for the sorted file should be adequate for most applications, but it may be changed, as shown below.

Example

```
//stepname EXEC EDSSORT,NAME='userid.dsname',  
//          NEWNAME='userid.newdsname',SIZE=primary  
//EDSSORT.SYSIN DD *  
SORT  FIELDS=(25,3,A,5,2,D),FORMAT=CH
```

4.13.5 EDSSAVE

EDSSAVE copies a sequential data set from disk to tape, converting the output from edit to non-edit format.

Required Symbolic Parameters

NAME	Names the input data set; also names the output data set unless NEWNAME is specified.
RETPD	Specifies the number of days the data set on the output tape is to be kept.

Optional Symbolic Parameters

DISK	Supplies the input volume serial number. Required if the data set is not cataloged.
TAPE	Supplies the output volume serial number if a specific volser is required.
NEWNAME	Names the output data set.
STORAGE	Gives a unit name for input. Required if the data set is not cataloged.
STATUS	Specifies whether the output data set is OLD or NEW; NEW is the default.
DRIVE	Gives a unit name for output; the default is TAPE.
DEN	Specifies a code (3 or 4) to indicate the density for a 9-track tape.
SEQ	Gives the sequence number for the tape data set; the default is 1. Overwriting data set 1 destroys any subsequent data sets.
LBL	Defines the type of label on the output tape. SL (standard-labeled) is the default. NL (unlabeled) may be specified.
LRECL	Specifies a logical record length for the output data set. The default is 80; longer records will be truncated unless LRECL is given.
BLKSIZE	Specifies a block size for the output data set. The default is 11440.
RECFM	Specifies a record format for the output data set. The default is FB.
STEPEND	Indicates the final (end of step) disposition for the output data set; KEEP is the default.

OPTIONS Many of the options which may be specified for the output data set are documented in section 1 and are fully described in the *Edit Format Utility Package* manual. Two of the options are LIST, to create a listing of the input data set and NUMBERED, to place the line numbers in columns 73-80.

UTILOPT Specifies parameters passed to the EDSUTIL program to affect message and ABEND processing. For further information, consult the *Edit Format Utility Package* manual.

The next example will save a cataloged edit format data set from a disk volume to tape. Since all defaults are being used, the output data set will be written on a virtual tape in CARD format. The tape will be kept for 30 days.

Example

```
//stepname EXEC EDSSAVE,  
// NAME='userid.dsname',RETPD=30
```

The following example saves a cataloged edit format data set from a disk volume to a foreign 3490 cartridge tape. Default values for the logical record length and block size of the output data set are being overridden.

Example

```
//stepname EXEC EDSSAVE,NAME='userid.dsname',  
// TAPE=tapeser,LRECL=133,BLKSIZE=11438,  
// DRIVE=FRGN3490  
//OUT DD EXPDT=98000
```

The following example saves a cataloged edit format data set from a disk volume to a virtual tape, requesting the placement of line numbers in columns 73 through 80 of each logical record. The output data set will be renamed. The tape will be kept for 7 days.

Example

```
//stepname EXEC EDSSAVE,NAME='userid.dsname1',  
// OPTIONS=NUMBERED,  
// NEWNAME='userid.dsname2',REPTD=7
```

4.13.6 EDSGET

EDSGET converts a non-edit format data set on tape to edit format and retrieves it to disk.

Required Symbolic Parameters

NAME Names the input data set; also names the output data set unless NEWNAME is specified.

Optional Symbolic Parameters

NEWNAME Names the output data set.

TAPE Supplies the input volume serial number. Required if the data set is not cataloged.

STORAGE Gives a unit name for output; the default is LONGTERM.

STATUS Specifies whether the output data set is OLD or NEW; NEW is the default.

DRIVE Gives a unit name for input; the default is TAPE.

DEN Specifies a code (3 or 4) to indicate the density for a 9-track tape.

SEQ Gives the sequence number for the tape data set; the default is 1.

UNITS Gives the units of allocation for the SPACE parameter; the default is TRK (tracks).

SIZE Gives the primary space allocation for the disk data set; the default is 10.

INCR Gives the number of units in each secondary allocation; the default is 5.

UNUSED Determines whether unused space is released; the default is RLSE. Code UNUSED= to retain unused space.

LBL Defines the type of label on the tape. SL (standard-labeled) is the default. NL (unlabeled) may be specified, but DCB information must be supplied if the input tape is NL.

LRECL Gives the logical record length for the input data set; it is required if LBL=NL and should not be used for standard-labeled tapes.

BLKSIZE Gives the block size for the input data set; it is required if LBL=NL and should not be used for standard-labeled tapes.

RECFM Gives the record format for the input data set; it is required if

LBL=NL and should not be used for standard-labeled tapes.

- STEPEND** Indicates the final (end of step) disposition for the output data set; CATLG is the default.
- OPTIONS** Many of the options which may be specified for the input data set are documented in section 1 and are fully described in the *Edit Format Utility Package* manual. Two of the options are LIST, to create a listing of the input data set and NUMBERED, to obtain the line numbers from columns 73-80 of the input data set.
- UTILOPT** Specifies parameters passed to the EDSUTIL program to affect message and ABEND processing. For further information, consult the *Edit Format Utility Package* manual.
- DISK** Supplies the output volume serial number. Required if the data set is to be written to a dedicated disk.

The following example retrieves a non-edit format data set from a cartridge tape to a disk volume. The output data set will be in edit format and will be cataloged and saved in the NOBACKUP management class.

Example

```
//stepname EXEC EDSGET,NAME='userid.dsname',  
// TAPE=tapeser,STORAGE=NOBACKUP
```

The following example retrieves a non-edit format data set from a foreign 3490 cartridge tape received from another installation to a cataloged data set in the LONGTERM management class. The output data set will be in edit format and will be renamed and cataloged.

Example

```
//stepname EXEC EDSGET,NAME='dsname1',  
// TAPE=tapeser,DRIVE=FRGN3490,  
// NEWNAME='userid.dsname2'  
//IN DD EXPDT=98000
```

The following example retrieves a non-edit format data set from a cartridge tape to a disk data set in the TEMP management class. The output data set will be in edit format and will be renamed. The default space allocation for the output data set will be overridden; unused space will be retained.

Example

```
//stepname EXEC EDSGET,
```

```
// NAME='userid.dsname1',STORAGE=TEMP,  
// TAPE=tapeser,NEWNAME='userid.dsname2',  
// SIZE=primary,UNUSED=
```

4.14 Generation Data Sets

An excellent explanation of how to use generation data sets can be found in Appendix B of the manual *OS/390 MVS JCL User's Guide*, GC28-1758. For more advanced information, consult the *IBM DFSMS/MVS Using Data Sets*, SC26-4922 and *IBM DFSMS/MVS Access Method Services for the Integrated Catalog Facility*, SC26-4906.

Briefly, a generation data set is one of a group of data sets cataloged chronologically. A generation data group (GDG) is a collection of generation data sets. Numbers are used to indicate the relative "age" of a data set. The data sets may have like or unlike DCB attributes and data set organizations. If the DCBs and organizations of all generations in a group are identical, they can be retrieved as one data set. Separate generations are referred to by coding the name of the generation group followed by a number in parentheses. A "+" in front of the number indicates a new generation added within the job. A "-" in front of the number indicates an older generation. If the number is a zero (with no sign), it indicates the current generation.

The generation numbers are updated at the end of the job. Therefore, generation data sets created and cataloged in a job are retrieved by their relative generation data set numbers throughout that job. (i.e., generation data sets +1 and +2 must be retrieved as +1 and +2 during that job.) If the job abnormally terminates and must be restarted, generation data sets that were created and cataloged before the restart step must have their relative generation data set numbers updated before the restarted job can be submitted. (i.e., generation data sets +1 and +2 created before the ABEND, must be referred to as -1 and 0 in the resubmitted job.)

Generation data sets may reside on disk or tape. All creation, deletion and modification of GDG members and their respective catalog entries should be performed by batch jobs. Any attempt to do otherwise, e.g., a WYLBUR or TSO command, can corrupt the structure and chronology of the entire GDG.

When a new generation is added to a generation data group, the data set name for the new generation must be entered in the catalog. This means that DISP=(NEW,CATLG) must be coded on the DD statement. If this is not done, duplicate data set names will be created in subsequent jobs.

The number of generations that are cataloged is determined by a number supplied when the generation data group index is created. When that number is reached, all the entries are uncataloged or the oldest entries only are either uncataloged or deleted. This option is one of those supplied when the generation index is created.

A model data set label or a data set used for this purpose must be created before a generation data group can be used.

4.14.1 The Model Data Set Label

A data set which is referenced by the operating system to obtain DCB attributes for new GDG data sets must exist before a new GDG data set is created. This data set, used as the model data set label, must be cataloged and must reside on a direct access volume. The data set name cannot contain special characters (except for periods used in a qualified name) and cannot be a generation data group name. The DCB parameter of the DD statement which describes the new GDG data set references the data set being used as the model.

If a label contains all the attributes required to define the new generation data set, only the name of the data set need be coded, (e.g., DCB=userid.dsname).

If all the attributes required to define the new generation data set are not contained in the label, or if certain ones are to be overridden, follow the data set name with these attributes, (e.g., DCB=(userid.dsname,list-of-attributes)).

4.14.2 The Cataloged Index

After the model data set label has been created, a job must be run using the VDSUTIL procedure to build the index for the generation data group and define what action should be taken when the index overflows. The VDSUTIL procedure invokes Access Method Services (AMS); VDSUTIL and AMS are described in the manuals *Using VSAM and AMS at NIH, DFSMS/MVS Access Method Services (ICF)*, SC26-4906, and *DFSMS/MVS Using Data Sets*, SC26-4922.

The following example will create a generation data group index, with 3 entries, and delete anything older than 3 generations.

Example

```
//STEPX EXEC VDSUTIL
//SYSIN DD *
DEFINE GDG (NAME(userid.NEWGDG.BASE) LIMIT(3) SCRATCH)
/*
```

SCRATCH should always be specified, otherwise the oldest version of the GDG will only be uncataloged, not scratched, and will continue to incur storage costs. LIMIT is a required parameter and must be specified.

The following example changes the number of generations for an existing GDG by setting a new LIMIT for the GDG.

Example

```
//STEPY EXEC VDSUTIL
//SYSIN DD *
ALTER userid.NEWGDG.BASE LIMIT(6)
/*
```

4.14.3 Creating a New Generation Data Set

Once the data set used as the model data set label and the cataloged index have been established, the generation data sets may be created.

The following DD statement describes a generation data set named `userid.UPDTE` that will be written to a data set in the `DISK2YR` management class. The data set `userid.LABEL` is used as the model.

Example

```
//ddname DD DSN=userid.UPDTE(+1),
// DCB=userid.LABEL,UNIT=DISK2YR,SPACE=(CYL,(5,2),RLSE),
// DISP=(NEW,CATLG,DELETE)
```

The following DD statement describes a generation data set named `userid.QTRLY` which will be written to a foreign 3490 cartridge tape and sent to another installation. The data set `userid.LABEL` is used as the model.

Example

```
//ddname DD DSN=userid.QTRLY(+1),
// DCB=(userid.LABEL),UNIT=FRGN3490,
// DISP=(NEW,CATLG,DELETE),VOL=(PRIVATE,SER=tapeser)
```

4.14.4 Deleting a Generation Data Group

The procedure `VDSUTIL` is used to scratch and uncatalog generation data groups. The `AMS DELETE` command indicates the name of the generation data group.

In the following example, `userid.NEWGDG.BASE` is the name of the generation data group. This procedure will delete the GDG base index entry. All generations under this index entry must have been deleted before the index entry can be deleted.

Example

```
//STEPZ EXEC VDSUTIL
//SYSIN DD *
DELETE userid.NEWGDG.BASE GDG
/*
```



4.15 Partitioned Data Sets

This section describes the procedures used in creating and handling partitioned data sets. They are listed in the chart below.

Procedure Name	Section	Function
PDSALLOC	4.15.1	Allocates disk space for a PDS.
PDSDEL	4.15.2	Deletes one or more members from a PDS.
MEMADD	4.15.3	Adds or replace a member to a PDS (source, macro, or private procedure library).
USRPDSL	4.15.4	Prints contents of PDS members or member names only, with space and directory blocks allocated and used. No editing allowed. Lists names only or names and contents of members of a PDS.
PDSPACK	4.15.5	Condenses a PDS.
PDSCOPY	4.15.6	Copies entire PDS from disk to disk.
USRPDSC	4.15.7	Copies one or more members of one PDS to another.
PDSSAVE	4.15.8	Saves (unloads) a PDS on disk to tape as a sequential data set.
PDSGET	4.15.9	Retrieves (reloads) a sequential data set from tape and write it to disk as a PDS.
PDSLST	4.15.10	Prints the contents of entire PDS or selected members of PDS or lists the names of members. Can copy, extract portions of, list, and change the DCB characteristics of members. Can print characters with hex equivalent below.

PDS Handling can also be done using ADSRECOV (see section 5.3.3), DSSCR (see section 4.10.1), DSRENAME (see section 4.9.1), and COPY (see section 4.5.1).

4.15.1 PDSALLOC

PDSALLOC allocates disk space for a partitioned data set on a specific volume. The first step deletes the data set if it exists (if it does not exist, no action is taken); the second step allocates space for the new data set. The stepnames are SCRATCH and ALLOC.

Required Symbolic Parameters

NAME Names the partitioned data set.

Optional Symbolic Parameters

STORAGE Gives a unit name; the default is LONGTERM.

LRECL	Specifies a logical record length.
BLKSIZE	Specifies a block size.
RECFM	Specifies a record format.
UNITS	Gives the units of allocation for the SPACE parameter; the default is TRK.
SIZE	Gives the primary allocation for the SPACE parameter; the default is 20.
INCR	Gives the secondary allocation for the SPACE parameter; the default is 10.
INDEX	Gives the number of directory blocks for the SPACE parameter; the default is 5. Cannot be increased after the PDS is created. Each block can hold entries for 4 to 5 members.
STEPEND	Indicates a disposition for the data set; CATLG is the default.
DISK	Supplies the volume serial number. Required if the PDS is to be stored on a dedicated disk.

The following example allocates a cataloged partitioned data set in the LONGTERM management class, accepting all defaults:

Example

```
//stepname EXEC PDSALLOC,NAME='userid.dsname'
```

The following example allocates a cataloged partitioned data set in the LONGTERM management class, changing the values for units and primary and secondary space, and giving DCB values:

Example

```
//stepname EXEC PDSALLOC,UNITS=CYL,SIZE=10,
// INCR=2,NAME='userid.dsname',
// LRECL=80,BLKSIZE=11440,RECFM=FB
```

4.15.2 PDSDEL

PDSDEL deletes one or more members from a partitioned data set. The PDS is not deleted. To delete an entire PDS, use the DSSCR procedure. The space the member took up is not available for re-use until after the data set is condensed.

Required Symbolic Parameters

NAME Names the partitioned data set.

PROGRAM Names the members to be deleted; cannot exceed 100 characters. The members need not be programs.

Optional Symbolic Parameters

DISK Supplies the volume serial number. Required if the PDS is not cataloged.

STORAGE Gives the unit name. Required if the PDS is not cataloged.

The following example deletes a single member from a cataloged PDS.

Example

```
//stepname EXEC PDSDEL,NAME='userid.dsname',  
// PROGRAM=memname
```

The following example deletes several members from a cataloged PDS.

Example

```
//stepname EXEC PDSDEL,NAME='userid.dsname',  
// PROGRAM='memname1, memname2, memname3...'
```

4.15.3 MEMADD

MEMADD adds members to a partitioned data set, using the IBM utility IEBUPDTE. The PDS would normally be a source library, macro library, or private procedure library. Note that IEBUPDTE cannot handle members in load module form.

Because JCL conversion (which includes expanding procedures) is done as soon as a job enters the system rather than just before execution, /*BEFORE and /*AFTER cannot be used to separate jobs which will update and then test a procedure in a user's private library. The job to test the procedure must not be submitted until the update job has been run and the new version of the procedure is available.

Required Symbolic Parameters

NAME Names the partitioned data set.

Optional Symbolic Parameters

DISK Supplies the volume serial number of the disk where the PDS is

	stored. Required if the PDS is not cataloged.
STORAGE	Gives a unit name. Required if the PDS is not cataloged.
RECFM	Gives the record format of the PDS; required if it is not already defined.
LRECL	Gives the logical record length of the PDS; required if it is not already defined.
BLKSIZE	Gives the block size of the PDS; required if it is not already defined.
OPTIONS	Requests any PARM field options for IEBUPDTE. NEW is the default.
CORE	Specifies a region size. The default is 4096K.

A SYSIN DD statement must be included to specify the source of input. This would normally be specified as DD * or DD DATA (if the member contains JCL). The input normally consists of IEBUPDTE ADD statements each of which is followed by the complete text of a member. If //SYSIN DD DATA is used, no /* statements can be included in the data and the input must be terminated with a /* statement. The format of an ADD statement is:

```
./ ADD LIST=ALL,NAME=member
```

where 'member' is the name of the member to be added. The ./ must be in columns 1 and 2.

The text of the member must immediately follow the ADD statement. If the member already exists, it will be replaced (unless the OPTIONS parameter is used to specify otherwise). Other IEBUPDTE control statements may be used if desired.

The following example shows how MEMADD may be used to place JCL procedures into an existing cataloged PDS:

Example

```
// EXEC MEMADD,NAME='userid.LIB1'
//SYSIN DD DATA
./ ADD LIST=ALL,NAME=PROC1
    .
    .
    PROC1
    .
    .
./ ADD LIST=ALL,NAME=NEWPROC
```

```
      .  
      .  
      NEWPROC  
      .  
      .  
/*
```

4.15.4 USRPDSL

USRPDSL lists the directory and contents of members in a partitioned data set. It can also list just the directory. It is an NIH-written utility. USRPDSL can handle non-edit format data sets with variable (RECFM=V or RECFM=VB) or fixed length records. The maximum length that will be printed is 121 characters; any record greater than this length will be truncated.

Required Symbolic Parameters

NAME Names the partitioned data set.

Optional Symbolic Parameters

DISK Supplies the volume serial number. Required if the PDS is not cataloged.

STORAGE Gives the unit name. Required if the PDS is not cataloged.

OPTIONS NOLIST causes only the members' names to be listed instead of the names and contents.

This example will produce a printout showing the contents within each member of the PDS userid.MYSORSE.

Example

```
//stepname EXEC USRPDSL,  
// NAME='userid.MYSORSE'
```

The following example will produce a list of members in the PDS userid.SRCEPROG.

Example

```
//stepname EXEC USRPDSL,OPTIONS=NOLIST,  
// NAME='userid.SRCEPROG'
```

4.15.5 PDSPACK

PDSPACK condenses a partitioned data set. Condensing is necessary because otherwise the space previously occupied by a replaced or deleted member is not available. It is wise to back up the data set on disk (with PDSCOPY) or on tape (with PDSSAVE) before beginning the PACK procedure because, if the PDSPACK fails during the condense, the members in the data set are usually destroyed.

Required Symbolic Parameters

NAME Names the partitioned data set to be condensed.

Optional Symbolic Parameters

DISK Supplies the volume serial number. Required if the PDS is not cataloged.

STORAGE Gives the unit name. Required if the PDS is not cataloged.

UNITS Gives the units of allocation for the SPACE parameter; the default is TRK (tracks).

UNUSED Determines whether unused space is released; the default is RLSE. Code UNUSED= to retain unused space.

This example condenses a cataloged partitioned data set.

Example

```
//stepname EXEC PDSPACK,NAME='userid.dsname'
```

The UNUSED option releases the unused space and reduces the amount of space for which the user is charged.

This example condenses a cataloged partitioned data set retaining unused space.

Example

```
//stepname EXEC PDSPACK,NAME='userid.dsname',  
//      UNUSED=
```

If later the user needs additional space for expansion of the PDS, the original secondary space allocation will be used. Note that RLSE releases all unused space, and a data set can end up with less space than was originally allocated in the primary quantity.

4.15.6 PDSCOPY

PDSCOPY copies an entire PDS from one direct access volume to another. Internally, the PDSCOPY procedure invokes the NIH-written program DSCOPYJ, which in turn links to the IBM utility program IEHMOVE.

The size of the output data set and the size of its directory can be changed when the PDS is copied. The size of the output data set is based on the size of the input data set unless a non-default value is specified for one of the symbolic parameters used to determine size.

The IBM utility program IEHMOVE is old and out of date. IBM provides it for the sake of compatibility only and recommends that you use IEBCOPY or DFSMSDss instead.

This procedure has a problem when the output data set is going to a public disk. Unless you override the space request, the step will get a condition code of 8 and you will see the following message in your output "IEH478I MOVE/COPY REQUEST IGNORED BECAUSE TARGET VOLUME(S) MANAGED BY THE STORAGE MANAGEMENT SUBSYSTEM." The cause of the problem is the inability of the IBM utility program IEHMOVE to dynamically allocate new data sets on system managed volumes. To circumvent this problem, override any one of the space parameters (i.e., UNITS=, SIZE=, INCR=, or DIR=) for the output data set to change it from its default value.

Required Symbolic Parameters

NAME Names the input data set; also names the output data set unless
NEWNAME is specified.

Optional Symbolic Parameters

INDISK Supplies the input volume serial number. Required if the data set is
not cataloged.

NEWNAME Names the output data set.

INSTOR Provides a unit name for input. Required if the input data set is not
cataloged.

OUTSTOR Provides a unit name for output; the default is LONGTERM.

UNUSED Determines whether unused space is released; the default is RLSE.
Code UNUSED= to retain unused space.

STATUS	Specifies whether the output data set is OLD or NEW; NEW is the default.
STEPEND	Indicates the final (end of step) disposition for the output data set; CATLG is the default.
Note: The size of the output data set is based on the size of the input data set unless a non-default value is specified for one of the following symbolic parameters, which are used to determine size. If a non-default value is given for any of these parameters, the default values will be used for any of the others that are not specified.	
UNITS	Gives the units of allocation for the SPACE parameter; the default is TRK (tracks).
SIZE	Specifies the number of units in the primary allocation; the default is 10.
INCR	Gives the number of units in each secondary allocation; the default is 5.
DIR	Gives the number of directory blocks to be allocated; the default is 5.
OPTIONS	OPTIONS=NEWSPACE sets the size of the output data set to the default values: UNITS=TRK SIZE=10 INCR=5 DIR=5
OUTDISK	Supplies the output volume serial number. Required if the PDS is to be written to a dedicated volume.

The following example will copy an entire cataloged PDS and place the new, renamed copy in the TEMP management class and catalog it. The value of INCR is overridden so that the data set can be allocated on an SMS volume.

Example

```
//stepname EXEC PDSCOPY,NAME='userid.dsname1',
// OUTSTOR=TMP,NEWNAME='userid.dsname2',INCR=6
```

In the following example, a cataloged PDS will be copied to the dedicated disk named DOUG09. The new PDS will be renamed and cataloged.

Example

```
//stepname EXEC PDSCOPY,NAME='userid.dsname1',
// OUTSTOR=3390,OUTDISK=DOUG09,
// NEWNAME='userid.dsname2'
```

4.15.7 USRPDSC

USRPDSC copies one or more members from one partitioned data set to another. Internally, the USRPDSC procedure invokes an NIH-written program, which in turn links to the IBM utility program IEBCOPY. The output library must already exist. Members from the input data set will replace any identically-named members in the output data set.

Required Symbolic Parameters

INPUT Names the input data set from which members will be copied.

OUTPUT Names the output data set into which members will be copied.

Optional Symbolic Parameters

INDISK Supplies the input volume serial number. Required if the PDS is not cataloged.

OUTDISK Supplies the output volume serial number. Required if the PDS is not cataloged.

INUNIT Gives the unit name for the input data set. Required if the PDS is not cataloged.

OUTUNIT Gives the unit name for the output data set. Required if the PDS is not cataloged.

OPTIONS ALL indicates that all members of the input data set are to be copied.

If OPTIONS=ALL is not used, control statements are used to indicate which members are to be copied. The control statements must start in column 1 and contain the names of the members the user wishes to copy; each separated by a comma. Columns 1-66 may be used. A member name may not be split between control statements, but as many control statements as necessary may be used to contain all the members.

The following example copies members PROG1, SUB1, SUB2, SUB3, from the cataloged PDS named userid.TESTLIB into the cataloged PDS named userid.LOADLIB.

Example

```
//stepname EXEC USRPDSC,INPUT='userid.TESTLIB',  
// OUTPUT='userid.LOADLIB'  
//COPY.SYSIN DD *  
PROG1,SUB1,SUB2,SUB3
```

The following example copies all members from userid.LIB to the cataloged PDS named userid.PROGRAMS.

Example

```
//stepname EXEC USRPDSC,INPUT='userid.LIB',  
// OUTPUT='userid.PROGRAMS',OPTIONS=ALL
```

4.15.8 PDSSAVE

PDSSAVE unloads a partitioned data set from disk to tape.

Internally, the PDSSAVE procedure invokes the NIH-written program DSCOPYJ, which in turn links to the IBM utility program IEHMOVE.

Required Symbolic Parameters

NAME Names the input data set; also names the output data set unless **NEWNAME** is specified.

Optional Symbolic Parameters

DISK Supplies the input volume serial number. Required if the data set is not cataloged.

TAPE Supplies the output volume serial number. The default is to use a scratch tape.

NEWNAME Names the output data set.

STORAGE Gives a unit name for input. Required if the PDS is not cataloged.

STATUS Specifies whether the output data set is OLD or NEW; NEW is the default.

DRIVE Gives a unit name for output; the default is TAPE.

DEN Specifies a code (3 or 4) to indicate the density for a 9-track tape.

RETPD	Specifies the number of days to retain the data set on tape. Required unless overwriting an existing data set.
SEQ	Gives the sequence number for the tape data set; the default is 1. Overwriting data set 1 destroys any subsequent data sets.
LBL	Defines the type of label on the output tape. SL (standard-labeled) is the default. NL (unlabeled) may be specified for foreign tapes.
STEPEND	Indicates the final (end of step) disposition for the output data set; KEEP is the default.

The following example saves a cataloged PDS from a disk volume to a virtual scratch tape that is kept for 8 days.

Example

```
//stepname EXEC PDSSAVE,NAME='userid.dsname',
// RETPD=8
```

The following example saves a PDS to a foreign 3490 cartridge tape and renames it.

Example

```
//stepname EXEC PDSSAVE,NAME='userid.dsname1',
// TAPE=tapeser,DRIVE=FRGN3490,
// NEWNAME='userid.dsname2'
//OUT DD EXPDT=98000
```

4.15.9 PDSGET

PDSGET reloads a partitioned data set from tape to disk. Internally, the PDSGET procedure invokes the NIH-written program DSCOPYJ, which in turn links to the IBM utility program IEHMOVE.

The size of the output data set (primary allocation, directory block, etc.) can be increased when the PDS is reloaded. The size of the output data set is based on the size of the input data set unless a non-default value is specified for one of the symbolic parameters used to determine size.

The IBM utility program IEHMOVE is old and out of date. IBM provides it for the sake of compatibility only and recommends that you use IEBCOPY or DFSMSdss instead.

This procedure has a problem when the output data set is going to a public disk. Unless you override the space request, the step will get a condition code of 8 and you will see the following message in your output "IEH478I MOVE/COPY REQUEST IGNORED BECAUSE TARGET VOLUME(S) MANAGED BY THE STORAGE MANAGEMENT SUBSYSTEM." The cause of the problem is the inability of the IBM utility program IEHMOVE to dynamically allocate new data sets on system managed volumes. To circumvent this problem, override any one of the space parameters (i.e., UNITS=, SIZE=, INCR=, or DIR=) for the output data set to change it from its default value.

Required Symbolic Parameters

NAME Names the input data set; also names the output data set unless **NEWNAME** is specified.

Optional Symbolic Parameters

NEWNAME Names the output data set.

TAPE Supplies the input volume serial number. Required if the input data set is not cataloged.

STORAGE Gives a unit name for output; the default is **LONGTERM**.

STATUS Specifies whether the output data set is **OLD** or **NEW**; **NEW** is the default.

DRIVE Gives a unit name for input; the default is **TAPE**.

DEN Specifies a code (3 or 4) to indicate the density for a 9-track tape.

SEQ Gives the sequence number for the tape data set; the default is 1.

LBL Defines the type of label on the tape. **SL** (standard-labeled) is the default. **NL** (unlabeled) may be specified.

UNUSED Determines whether unused space is released; the default is **RLSE**. Code **UNUSED=** to retain unused space.

STEPEND Indicates the final (end of step) disposition for the output data set;

CATLG is the default.

DISK Supplies the output volume serial number. Required if the data set is to be written to a dedicated disk.

Note: The size of the output data set is based on the size of the input data set unless a non-default value is specified for one of the following symbolic parameters, which are used to determine size. If a non-default value is given for any of these parameters, the default values will be used for any of the others that are not specified.

UNITS Gives the units of allocation for the SPACE parameter; the default is TRK (tracks).

SIZE Specifies the number of units in the primary allocation; the default is 10.

INCR Gives the number of units in each secondary allocation; the default is 5.

DIR Gives the number of directory blocks to be allocated; the default is 5.

OPTIONS OPTIONS=NEWSPACE sets the size of the output data set to the default values: UNITS=TRK, SIZE=10, INCR=5, and DIR=5.

The following example retrieves a PDS from a virtual tape to a data set in the TEMP management class. The new data set will be cataloged and will be given the same name as the tape data set.

Example

```
//stepname EXEC PDSGET,NAME='userid.dsname',  
// STORAGE=TEMP,TAPE=tapeser,INCR=6
```

The following example will retrieve a PDS from a 3480 cartridge tape received from another installation to a cataloged data set in the LONGTERM management class. The output data set will be renamed.

Example

```
//stepname EXEC PDSGET,NAME='dsname1',INCR=6,  
// TAPE=tapeser,DRIVE=FRGN3480,NEWNAME='userid.dsname2'  
// IN DD EXPDT=98000
```

4.15.10PDSLST

PDSLST lists the contents of all members or of a selected member of a partitioned data set, using the LISTPDS program. It also lists the member names and any alias

names in a table of contents, along with the page number for the data. An alias in the table of contents has an "-A" designation following its name.

The PDS can be either edit or non-edit format. If the PDS has the DCB attributes of RECFM=U and a BLKSIZE greater than or equal to 1022, it is assumed to be edit format. The output formats are of two basic kinds: single-line and multi-line.

Each of the single-line formats lists only a single output line corresponding to each input logical record. The use of machine or ISO/ANSI carriage controls in the input data can be detected by PDSLIST if requested. If this option is not selected, spacing is controlled by the presence or absence of a carriage control indication (machine or ISO/ANSI) in the data set label or in the JCL. Data with carriage controls (as determined by either of the above methods) will have carriage spacing controlled by these characters; otherwise, single spacing will be used with continued headers between pages.

Each of the multi-line formats lists input records in their entirety by printing the record in 100-character segments. Multi-line formats also have continued headers between pages; double or triple spacing is provided between records; and any carriage controls are listed in the displayed data.

PDSLIST is controlled by the OPTIONS symbolic parameter. Several of the control keywords reference the line number field of input records (defined as the last eight data positions for fixed-format input records and the first eight data positions for U or V-format input records).

If you scratch a member, the member name is removed from the directory, but the data is not removed until you condense the PDS. However, if you have not removed the aliases for that member, condensing the PDS does not remove the data. In that case, the aliases are listed in the PDSLIST table of contents along with the page number for the data.

The PRINT utility described in section 4.7.2 can be used to list a single member of a PDS.

Required Symbolic Parameters

NAME Names the data set to be listed.

Optional Symbolic Parameters

DISK Specifies the volume serial number of the disk on which the PDS is stored. Required if the data set is not cataloged.

STORAGE Gives the unit name for the input device. Required if the PDS is not cataloged.

OPTIONS Selects options. Options are listed below under "Option List." If a control keyword is coded more than once, only the last one will affect the program (except that the others must also be syntactically correct). If this symbolic parameter is not coded, then the default options will be `FORMAT=LIST`, `MAX=9999999`, `PAGE=60`, `SKIP=60`.

Option List

`FORMAT=format`

Specifies which one of several output formats is desired. The `LIST`, `AORM`, `SNUM`, `NUM` and `NOFORM` formats each output a single line for each input record; the `MULTI`, `HEX` and `DEBUG` formats produce as many output lines as required to list each input record fully.

- `FORMAT=LIST` results in the **LIST**ing and examining of the first record of each member. If the line number field of the first record is numeric, the member will be printed with `NUM` format; otherwise, `NOFORM` format will be used.
- `FORMAT=AORM` determines if both **ISO/ANSI OR Machine** carriage control is to be used and examines the line number field. Any carriage control indication from the label or the `JCL` is ignored.
- `FORMAT=SNUM` Suppresses **NUM**bers (i.e., line numbers) field of each line.
- `FORMAT=NUM` requests that the line **NUM**ber for each output line be formatted as follows: line number field (with any leading zeroes suppressed), a blank, and the data portion.
- `FORMAT=NOFORM` requests **NO FORM**ating changes to the data lines being listed.
- `FORMAT=MULTI` requests **MULTI**-line format, that is, that each input record be printed out fully with 100 characters per output line.
- `FORMAT=HEX` requests **HEX**adecimal format, that is, that each input record be printed out fully in a hexadecimal format that uses two output lines for each 100 characters (first line for the upper nibble of each byte, second line for the lower nibble of each byte).
- `FORMAT=DEBUG` requests that each input record be printed out fully in a combined `MULTI` and `HEX` format.

MAX=number

Specifies the maximum number of input records that are to be processed for any member. For example:

- MAX=9999999 requests that at most 9,999,999 input logical records be listed for any member.
- MAX=4 requests that only the first four input logical records be listed for any member.

PAGE=number

Specifies the length (in lines) of an output page. For example:

- PAGE=60 requests that pages contain no more than 60 lines.
- PAGE=30 requests that pages contain no more than 30 lines.

SKIP=number

Specifies the number of blank lines that are to be listed between members in the output. Note that upon reaching the top of a following page, no additional blank lines will be produced. For example:

- SKIP=60 requests that the listing of each member begin on a new page.
- SKIP=6 requests that six blank lines be printed between members in the output listing.

The syntax scanner that processes the options list is simple. It uses only the first two characters of any keyword and the first two characters of FORMAT operands. Commas or blanks may be coded as separator characters between the keyword entries. Thus, OPTIONS='FO=LI,PA=34' is entirely equivalent to

```
OPTIONS= '  ,FORMAT=LIST    PAGE=34    ' .
```

The following example will list a cataloged PDS.

Example

```
//stepname EXEC PDSLIST,NAME='userid.dsname'
```

The following example will list a single member of a PDS from a cataloged data set using ISO/ANSI carriage control.

Example

```
//stepname EXEC PDSLIST,NAME='userid.dsname(memname) ',  
//    OPTIONS='FORMAT=AORM'
```

4.16 VSAM Data Sets

Virtual storage access method (VSAM) data sets can be processed using Access Method Services, which are invoked in batch mode by executing PGM=IDCAMS. See the VDSUTIL procedure in Section 4.3.1 for an easy way to execute IDCAMS. The most commonly used functions for Titan are listed below:

- defining the base catalog entry for a Generation Data Group
- defining VSAM data sets
- modifying VSAM data set attributes
- loading/unloading VSAM data set data
- defining and building alternate indexes for VSAM data sets
- deleting VSAM data sets
- printing VSAM data sets
- checking the structural integrity of VSAM data sets and indexes

For addition information about IDCAMS, refer to IBM's manual *DFSMS/MVS Access Method Services for Integrated Catalog Facility*, SC26-4906.

4.17 Using ISMF

ISMF (the Interactive Storage Management Facility), enables you to manage your disk data sets interactively. You can use ISMF to recover a data set from an incremental backup, alter the management class, delete a backup version, or create a new backup version.

4.17.1 Accessing ISMF

To access ISMF from the main ISPF menu:

1. Select C (Products)
2. Select IS (ISMF).

The ISMF Primary Option Menu is displayed.

You can use option 0 to modify your profile or option 1 to manage data sets. The other options are intended for the CIT storage administrators.

4.17.2 Specifying Data Set Names

Follow these guidelines to specify data set names:

For data sets owned by the logon userid:

To	Do	Example
Specify a data set	Use the simple dsname. Do not include your userid.	TEST1
Specify a wildcard to represent one character	Use %	TEST%
Specify a wildcard to represent one qualifier (group of characters between dots) in addition to the userid	Use *	*
Specify a wildcard to represent one or more qualifiers in addition to the userid	Use**	**

For data sets owned by another userid:

Specify a userid other than the logon userid	Enclose the complete dsname in single quotation marks	'userid.dsname' 'userid.TEST%' 'userid.*' 'userid.**'
--	---	--

4.17.3 Displaying Data Set Information

From the ISMF Primary Option Menu select 1 (Data Set) to perform functions against data sets.

1. On the Data Set Selection Entry Panel, enter the following:

- For Source of Generated List, enter 2

This generates a new list of data sets.

- For Data Set Name, specify the name using the guidelines above.
- For Source of the new list, enter 2.

This uses the catalog as the source.

-
- For Acquire Data from Volume, enter Y if you would like to view the space and DCB information for each data set.
 - For Acquire Data if DFSMSHsm Migrated, enter Y if you would like to view the management class for each data set.

2. Press Enter.

The results are displayed on the Data Set List panel.

4.17.4 Determining Total Space

1. To determine the total space allocated and used by the data sets in the current list, enter LISTPRT on the command line.

The Data Set Print Entry Panel is displayed.

2. To specify which columns of information you want to print, replace the * with the column numbers on the Specify Tags to be Printed line.

For example, entering “3 4 9 20” would cause the columns for allocated space, used space, unit of allocation, and creation date to be printed.

3. Press Enter.

The Print Job Submission Entry Panel is displayed.

At any time you can press PF1 to get more information.

4.17.5 Changing Management Class

You can change the management class for a data set. The management class is displayed in column 26.

1. On the Data Set List panel, enter ALTER in the Line Operator column for the data set that you want to modify.

2. On the Data Set Alter Entry Panel, change the management class.

4.17.6 Deleting Specific Backups

You can delete specific backup versions of a data set.

1. On the Data Set List panel, enter HBDELETE in the Line Operator column for the data set.

2. On the HBDELETE Entry Panel, enter Y for any backup that you want to delete.

4.17.7 Recovering a Data Set from a Backup

You can recover a data set from a backup. The data set must be in a management class that provides backup services.

There are two ways to recover the data set.

Method for Existing Data Sets

1. Display the data set in the Data Set List panel.
2. Enter the HRECOVER in the Line Operator column for the data set.
3. On the HRECOVER Entry Panel, enter Y for the backup that you want to recover.
4. On the next HRECOVER Entry Panel, specify whether to overwrite the existing data set or restore the data set to a new name.

Method for Deleted Data Sets

If you delete a data set that has backups, the backups are kept for six weeks. During that time, you can recover the data set.

1. Generate a list of data sets that have backups to make sure you have the exact name of the data set. Use the BACKLIST procedure (see section 4.19) to create the list.
2. Enter the HRECOVER command to recover the data set. You can enter the command from:
 - ISPF panel 6.
 - The READY prompt.
 - The command line of any ISPF panel. Precede the command with TSO. For example:

TSO HRECOVER ...

Required Parameters

NAME	Names the data set to be recovered.
FROMVOLUME(valid)	Required only if the data set is on a dedicated volume and uncataloged.

Optional Parameters

GENERATION(genum) or DATE(date) or VERSION(vernum)	The generation, date, or version you want to recover if not the most recent.
NEWNAME(newdsname)	A new name for the recovered data set
TOVOLUME(valid)	A volume to recover the data set to. The default is the volume of origin.
UNIT(unitname)	The unit name for the device to recover the data set to
WAIT or NOWAIT	WAIT specifies that you will wait for the recovery to complete. NOWAIT specifies that the recovery should continue in the background. The default is NOWAIT.

If the HRECOVER command ends with an error and RC=0168, specify the parameters TOVOLUME(DSP101) UNIT(3390) on the command.

Examples

```
HRECOVER dsname WAIT
```

Recovers the most recent backup of the data set you specify and waits for the recovery to finish. Only data sets under your userid are considered.

```
HRECOVER 'userid.dsname'
```

Recovers the most recent backup of a data set beginning with a userid other than the one you are logged on with.

4.18 Obtaining Data Set Information

4.18.1 ADSMAP

The ADSMAP procedure lists the disk data sets currently on public storage. The list includes data sets that are online or migrated by HSM. The data sets are sorted by name with each high-level qualifier starting a new page. The SOURCE column indicates where the data set was found:

- VTOC if online
- MCDS if migrated by HSM

ADSMAP executes the FDREPORT program from Innovation Data Processing.

WARNING: ADSMAP is not cheap. It costs about \$13 to run if the program scans every online volume. If you limit the search to fewer volumes, the cost goes down. It is less than \$3 for only one volume. Overall, it is cheaper to run ADSMAP once specifying all prefixes rather than to run ADSMAP separately for each prefix.

Required Symbolic Parameters

PREFIX Specifies one or more data set name prefixes, up to 44 characters each. If you specify multiple prefixes, they must be separated by commas. Any data set name that matches one of the prefixes will be listed.

Optional Symbolic Parameters

VOLUMES Specifies one or more online disk volume serial number prefixes to be searched for data sets. If you specify multiple prefixes, they must be separated by commas. Any online disk whose volume serial number matches one of these prefixes will be searched for data sets. The default is 'DS,DIS,PUB' which includes all online public disks.

The VOLUMES parameter limits only the online volumes. All migrated volumes are searched regardless of whether the VOLUMES parameter is used.

Example

The following example lists all data sets whose names begin with the string \$iii. Data sets on public volumes and migration volumes are listed.

```
//stepname EXEC ADSMAP,PREFIX='$iii'
```

Example

The following example lists all data sets whose names begin with the string \$iii, the string aaaaiii, or the string \$jjj.more. Data sets on public volumes and migration volumes are listed.

```
//stepname EXEC ADSMAP,PREFIX='$iii,aaaaiii,$jjj.more'
```

Example

The following example assumes that you have a number of dedicated disks whose volume serial numbers all begin with the letters ABC. This job lists all data sets on any of these disks whose names begin with aaaa. Migrated data sets are also listed even if they were migrated from other volumes.

```
//stepname EXEC ADSMAP,PREFIX='aaaa',  
// VOLUMES='ABC'
```

4.19 Obtaining List of Backups for Data Set

4.19.1 BACKLIST

The BACKLIST procedure lists the name, date, time, and volume for each backup of a data set. Titan keeps up to five backups, created when HSM runs its daily incremental backup, backing up only those data sets that have changed. BACKLIST executes the HSM HLIST command in batch.

Required Symbolic Parameter

PREFIX Specifies either a data set name prefix or a data set name. Any data set name that matches the prefix will be listed.

Optional Symbolic Parameter

OUTCLASS Specifies the SYSOUT output class that is to be used to print the listing. The default is 'H'.

Example

The following example lists backup information for all data sets whose names begin with the string \$iii. The period in the search string limits the search to a high-level qualifier that exactly matches the prefix. If you leave the period off, then information for data sets beginning with \$iiijj would also be listed.

```
//stepname EXEC BACKLIST,PREFIX='$iii.'
```

Example

The following example lists all data sets whose names begin with the string aaaaiii. The list will be printed to SYSOUT class A.

```
//stepname EXEC BACKLIST,PREFIX='aaaaiii.',  
// OUTCLASS=A
```

5 Disk Utilities

Disk volumes reserved for use of a particular user group must be maintained by their owner. Data sets on dedicated disks must follow standard naming conventions but do not have to be cataloged. We strongly recommend, however, that you catalog these data sets.

5.1 Guidelines for Maintaining Dedicated Disks

The following suggestions and guidelines are designed to assist users in the proper maintenance of their dedicated disks. Users are responsible for the integrity of the data on all disk volumes that are assigned to them.

5.1.1 Technical Coordinator

To avoid possible confusion, each group must appoint a technical coordinator to be responsible for the maintenance of the disk volume. This person is responsible for maintaining a current backup of each disk and will be contacted if any disk assigned to that group goes bad and causes problems for the central facility. The coordinator must be designated at the time the disk volume is requested. The coordinator must be able to contact all users of the disk quickly in case use must be suspended while recovery procedures are performed.

5.1.2 Backup Policy

Since the ONLY way to recover a disk volume is by restoring its contents from a good backup tape or, alternatively, recreating every data set on the disk, disks should be backed up to tape FREQUENTLY. The frequency of backups should be determined on the basis of how much activity there is against the disk volume. For example, if a disk is used by many people, is altered every day, or is accessed from TSO or WYLBUR, it should be mapped nightly and backed up at least twice weekly. All of the output should be carefully checked as soon as possible after a maintenance job is run.

Because magnetic tape does deteriorate with time, it is best to have at least four backup tapes, rotating among the tapes for each successive backup job.

If disk backups are not done nightly, active data sets should be backed up individually on a fixed schedule (preferably nightly) if they are vital to a project.

5.1.3 Emergency Procedures

If jobs accessing data sets on a dedicated disk volume ABEND with errors, such as "VTOC ERRORS MAY EXIST" or "DATA CHECKS," or if there are persistent I/O errors on any track on the disk, contact the technical coordinator of the disk immediately.

For any dedicated disk that is suspected of errors, the technical coordinator should see that all users of the disk SUSPEND ALL USAGE OF THE DISK. The coordinator should contact the NIH Help Desk as soon as possible (by submitting a Service

Request Ticket or by telephoning the NIH Help Desk. The staff will work with the dedicated disk technical coordinator to ascertain the nature of the problem, to determine the best solution to the problem, and to help the users get started using the disk again.

If the problem with the dedicated disk occurs outside of regular consulting hours and there is a time-critical need to use the disk, the technical coordinator for the dedicated disk can take the following steps:

- Suspend all usage of the disk.
- Call the Computer Operations shift coordinator (301-496-4715), describe the problem and the reasons for "outside of regular hours" action. The Shift Coordinator will work with the disk technical coordinator and will contact Computer Center systems support staff to arrive at the best solution to the problem (for example, initializing a new disk with the volume serial number for the purpose of a disk restore).
- The technical coordinator of the dedicated disk will be contacted by Computer Center staff when the disk problem has been resolved.
- Run the DISKMAP utility on the volume to verify that the correct number of tracks for the entire disk are accounted for.
- Restore the disk from the most recent backup tape. The DISKGET procedure can be used for this purpose.
- Verify that the volume has been restored by checking the output of the DISKMAP portion of the DISKGET job (or by running a separate DISKMAP job).

When the disk has been successfully restored, the technical coordinator should notify users of the disk that usage can resume. Users of the dedicated disk should also be notified whenever any data sets that were updated after the creation of the most recent backup tape will have to be recreated either by re-running the jobs that updated the data sets or by restoring the data sets from their own separate backups.

If Computer Center staff detects hardware errors with a dedicated disk, they will contact the dedicated disk technical coordinator. If necessary, the dedicated disk may be made unavailable to users until the disk coordinator has been contacted.

The procedures for dumping and restoring a dedicated disk volume are documented in section 5.3; the DISKMAP procedure is described in section 5.2.

5.1.4 Compressing a Disk

Even though a disk volume may have many tracks of available free space, it is possible that this free space will be "scattered about the disk" in many small pieces.

This situation is referred to as fragmentation. When a disk is highly fragmented, the disk coordinator may request a compress of the disk via a Service Request Ticket. A current DISKMAP of the disk must be submitted as documentation for the Service Request Ticket. Before requesting a compress, the coordinator should ensure that the space on the disk is appropriately utilized (e.g., old unused data sets have been deleted, etc.).

The compress is a free service, which is scheduled at an available nighttime or weekend opportunity. It is usually done between midnight and 7:00 a.m. Usage of the volume will have to be suspended for the time frame in which the compress is scheduled. It is advisable to back up the disk immediately after a successful compress.

5.1.5 Hints on Maintaining Dedicated Disks

Here are some for maintaining dedicated disks:

- Always check the output of the DISKMAP after each dump and restore job. If you are not executing a procedure, which includes a DISKMAP, include another step in your job to execute DISKMAP before each dump and after each restore.
- Always dump or restore your disks when there is no activity against them (e.g., at night using the LATE resource).
- Batch jobs accessing data sets on a dedicated volume should contain a `/*CNTL controlname,SHR` statement. Backup or restore jobs should contain a `/*CNTL controlname,EXC` statement. In both cases, the "controlname" should be the same.

If you require help and have followed the guidelines specified above, call the NIH Help Desk and be prepared to deliver a current DISKMAP run of the disk volume to the staff. The consulting staff will attempt to guide and advise you on the best and least expensive way to recover the disk.

5.2 Mapping Disks

5.2.1 DISKMAP

DISKMAP maps the volume table of contents (VTOC) of a disk. It executes the FDRABRP program from Innovation Data Processing.

The information it displays includes the data set name, date last referenced, data set organization, record format, block size, logical record length, tracks allocated, and tracks free.

Required Symbolic Parameter

DISK Supplies the volume serial number. If you want to map two disks, place the disk names in parentheses and separate them with commas.
Example:
 DISK=(volser1,volser2)

The following example will list the characteristics of a user's data sets on a private disk.

Example

```
//stepname EXEC DISKMAP,DISK=volser
```

5.3 Backing Up and Restoring Disks

The DISKSAVE and DISKGET procedures are available to back up and restore a dedicated (private) disk. Both use the FDR program. Only the technical coordinator of the dedicated disk has the RACF authority that will allow that person to successfully run the DISKSAVE, DISKGET, and ADSRECOV procedures. By using the ADSRECOV procedure, the backup tape created with DISKSAVE can be used to retrieve individual data sets (except VSAM data sets). For more information on ADSRECOV, see section 5.3.3.

The procedures DISKSAVE and DISKGET each execute a DISKMAP to map the disk volume before it is dumped and after it is restored, respectively. To ensure the integrity of the data, all activity against a disk must be suspended when it is being dumped to tape or restored from tape.

It is particularly important to check the output of a backup job to ensure that it completed properly. Check the JES2 Job Log and the completion code of each step for problems.

5.3.1 DISKSAVE

DISKSAVE creates an FDR full volume tape backup of a dedicated disk. It is a two-step procedure. The first step produces a listing showing all data sets stored on the specified disk using the same program that DISKMAP uses. The second step dumps the disk to tape. Only the technical coordinator of the dedicated disk has the RACF authority that will allow that person to successfully run the DISKSAVE procedure.

Required Symbolic Parameters

DISK Supplies the volume serial number of the user's dedicated disk to be backed up.

TAPE Supplies the 6-digit volume serial number for the output tape. Several tapes may be specified as a list of volume serial numbers separated

by commas and enclosed within parentheses inside single quotation marks.

NAME Names the output data set on tape.

RETPD Specifies the number of days to retain the data set on tape. Required unless overwriting an existing data set.

Optional Symbolic Parameters

STORAGE Gives the unit name for the dedicated disk: 3390 is the default.

DRIVE Gives the unit name for the output; the default is LTAPE.

DEN Specifies a code (3 or 4) to indicate the density for a 9-track tape.

STATUS Specifies whether the output data set on tape is OLD or NEW; NEW is the default.

STEPEND Indicates the final (end of step) disposition for the output data set on tape; KEEP is the default.

The following JCL will dump a dedicated disk onto a cartridge tape.

Example

```
//stepname EXEC DISKSAVE ,DISK=volser ,TAPE=tapeser ,  
// NAME= 'userid.dsname' ,RETPD=30
```

5.3.2 DISKGET

DISKGET restores the complete contents of a user's dedicated disk from an FDR full volume tape backup. It should not be used to restore a disk from a tape created at another installation and cannot be used to restore to a different model of disk. Only the technical coordinator of the dedicated disk has the RACF authority that will allow that person to successfully run the DISKGET procedure.

Required Symbolic Parameters

DISK Supplies the volume serial number of the disk to which the backup tape is being restored.

NAME Names the input data set on tape.

Optional Symbolic Parameters

TAPE Supplies the volume serial number of the backup tape. Several tapes may be specified as a list of volume serial numbers separated by

commas and enclosed within parentheses inside single quotation marks. If more than one tape is specified, they must be specified in the same order used when the backup was performed. Required if the input data set is not cataloged.

DRIVE Gives the unit name for the tape; the default is LTAPE.

The following JCL will restore the contents of a dedicated disk from two backup tapes.

Example

```
//stepname EXEC DISKGET,DISK=volser,  
//          NAME='userid.dsname',  
//          TAPE='(tapeser1,tapeser2)'
```

5.3.3 ADSRECOV

ADSRECOV recovers data sets from an FDR full volume tape backup of a dedicated disk. Data sets or VSAM clusters can be restored using DSF control statements. Data sets can be restored by specific name, by a high level prefix, or by using filter characters.

Only the technical coordinator of the dedicated disk has the RACF authority needed to run the ADSRECOV procedure.

If an existing data set is being overwritten using the same name or if the user allocates a data set with the old name, the technical coordinator does not need access to the data set. The UACC can be anything, even NONE.

If the data set is being restored using a new name, the technical coordinator must have ALTER authority to the new name. The owner can grant the technical coordinator ALTER authority.

In either case, the DCB characteristics of the data set are restored when the data set is restored.

Below are some other options when restoring data sets.

- Data sets being restored can be renamed while being restored.
- Data sets can be restored to a different volume. (normal restore is the same volume).
- Multi volume data sets can be restored. (only to the same number of volumes).
- FDRDSF supports SMS management of data sets.

- Restore NON-SMS data sets to SMS Volumes.

Required Symbolic Parameters:

DISK Supplies the volume serial number of the disk to which the backup tape is being restored.

NAME Names the input data set on tape.

Optional Symbolic Parameters:

TAPE Supplies the volume serial number of the backup tape. Several tapes may be specified as a list of volume serial numbers separated by commas and enclosed within parentheses inside single quotation marks. If more than one tape is specified, they must be specified in the same order used when the backup was performed. Required if the input data set is not cataloged.

DRIVE Gives the unit name for the tape; the default is LTAPE.

STORAGE Gives the unit name for the dedicated disk: 3390 is the default.

You need two control statements: RESTORE and SELECT.

The format of the RESTORE control statement is either

```
RESTORE        TYPE=DSF
```

or

```
RESTORE        TYPE=DSF , PRESTAGE
```

The PRESTAGE option prevents output data set that exist on the target volume from being restored. It may be used to avoid restoring data sets that have already been restored.

The format of the SELECT control statement is:

```
SELECT         DSN=dsname
```

or

```
SELECT         DSN=dsname , NEWNAME=newdsname
```

or

SELECT DSN=filter

You may use the following filter characters to specify the data sets to be selected:

- | | |
|---|--|
| / or % | represents ANY single valid alphanumeric or national character. |
| (vertical bar) | represents any single alphabetic (A-Z) character. |
| + | represents any single numeric (0-9) character. |
| ? | represents any single national (\$ # @) character. |
| . (period) | is used to separate index levels. For compatibility with earlier releases, if the filter begins with one or more periods, this indicates that that many index levels at the beginning of the name are to be skipped over before applying the rest of the filter. |
| * | by itself as an index level indicates that the index level must exist in the selected name, but that it can contain any valid characters and can be any valid length (1 to 8 characters). However, if a single asterisk is combined with other characters in an index level, then it represents a variable length string (zero or more characters) at that point in the index level. |
| ** (double asterisk with no trailing period) | represents zero or more characters or index levels. The filter routine will try to apply the remainder of the filter to the data set name beginning with the current character and stepping through the name until it matches or until the end of the name is reached. |
| ** . (double asterisk with a trailing period) | also represents zero or more characters or index levels, but the filter routine will try to apply the remainder of the filter to the data set name at the beginning of each index level (if not currently at the beginning of a level, it will start at the next level). |

These special cases apply to ** with periods:

- | | |
|-------------------------------------|--|
| ** . at the beginning of the filter | represents zero or more whole index levels at the beginning of the name. |
| .** at the end of the filter | represents zero or more whole index levels at the end of the name. |

The following JCL can be used to execute the ADSRECOV procedure to recover a single data set from the user's backup tape. Note the data set being restored from the back up tape is not currently on the target disk.

Example

Restore a single data set by its full qualified data set name.

```
//stepname EXEC ADSRECOV,DISK=volser,TAPE=tapeser,
//      NAME='userid.tapename'
//SYSIN DD *
        RESTORE TYPE=DSF
        SELECT DSN=userid.dsname
/*
```

Example

Restore all data sets by its high level prefix, from same back up tape..

```
//stepname EXEC ADSRECOV,DISK=volser,TAPE=tapeser,
//      NAME='userid.tapename'
//SYSIN DD *
        RESTORE TYPE=DSF,PRESTAGE
        SELECT DSN=userid.**
/*
```

Example

Restore old data sets to a new name.

```
//stepname EXEC ADSRECOV,DISK=volser,TAPE=tapeser,
//      NAME='userid.tapename'
//SYSIN DD *
        RESTORE TYPE=DSF,PRESTAGE
        SELECT
        DSN=userid.dsname,NEWNAME=userid.newdsname
/*
```

6 Tape Utilities

This section describes the TAPEINFO and TAPEMAP procedures and the TI command.

For the more information on tapes and tape utilities, see section 11, "Tape Storage," and the following link:

<http://silk.nih.gov/silk/tapes>

Tape handling can also be done using COPY (see section 4.5.1), PRINT (see section 4.7.1), COMPARE (see section 4.2), and SORT (see section 4.11.1).

The following table may aid the user in deciding which utility to choose depending upon the function to be performed. Where a choice of utilities is given, USE THE FIRST ONE LISTED unless a special feature of the other is required.

Functions of Tape Utilities

Function	Utility
Compare two data sets	COMPARE
Copy tape to tape	
SL to SL selected files	DSCOPY, COPY
SL to NL	DSCOPY, COPY
NL to SL	DSCOPY, COPY
NL to NL selected files	DSCOPY, COPY
AL to SL or NL	DSCOPY, COPY
Copy features:	
Reblock	COPY
Determine DEN for 9-track	TAPEMAP
Erase data on a tape	ADSERASE
Find out if a tape is labeled	TAPEMAP
Print	
Status information	TAPEINFO, TI
Tape labels	TAPEMAP
Data set labels	TAPEMAP
Specified file in hexadecimal	PRINT
Selected records	PRINT, EDSUTIL, EDSLIST
Specified files	DSLIST, PRINT
Begin at any record in any block	DSLIST, EDSUTIL, EDSLIST
Number of blocks and largest block size for each file	TAPEMAP
Put tape mark on new NL Special tape or re-label SL Special tape with new VOLSER	Output Distribution Services
Sort/Merge	SORT, MERGE
Update, replace or insert records	IEBUPDTE (tape to tape)

Notes on copying tapes:

- No more than two reel tapes may be mounted at one time.
- ASCII data may be written only on a user-supplied (foreign) 9-track tape.
- COPY - requires one step per data set to be copied. Cannot copy ASCII variable length data.

Other utilities available for saving and retrieving disk or tape data sets are described elsewhere in this section. See the *IBM DFSMS/MVS Utilities* manual, SC26-4926, for IEBUPDTE.

6.1 Displaying Tape Information

6.1.1 TAPEINFO

TAPEINFO prints status information that the CA-1 Tape Management System (TMS) has about a tape. It executes the TMSBINQ program from Computer Associates.

The information printed includes the data set name, expiration date, creation date, creation time, name of the job that created the tape, name of the program that created the tape, date last used, record format, record length, and block size. The information is gathered in real time and is therefore valid as of the time you run the procedure.

If the tape has multiple data sets on it, the status information for each data set is listed.

If the tape is part of a multi-volume data set, TAPEINFO lists the serial numbers of the first volume, next volume, and previous volume plus a sequence number indicating the location of this volume relative to the others. In addition, the status information for all volumes whose sequence numbers follow the specified volume is printed.

You need at least one control statement to tell TAPEINFO the tapes for which you want to print information. The format of the control statement is:

```
VOL=vvvvvv [-vvvvvv] [ , SHORT ] [ , NEWPAGE ]
```

Control Statement Parameters

VOL	Identifies the control statement. It must begin in column 1.
vvvvvv	Specifies the volume serial number of the tape. To specify a range of tapes, include the beginning and ending volume serial numbers separated by a dash. You can get the volume serial number of the tape by using the VOLLIST procedure (see section 6.1.2).

SHORT Specifies that only the data set name, expiration date, creation date, creation time, and name of the job that created the tape is printed for each tape. If this parameter is omitted, all information available is printed.

NEWPAGE Forces a page break between printouts when multiple VOL control statements are used. If NEWPAGE is not specified, no page break occurs when the next VOL control statement is processed. This parameter, if used, must be last.

Example

To print all the information available for tape 123456:

```
//stepname EXEC TAPEINFO  
VOL=123456
```

Example

To print the short list of information available for all tapes in the range 234560 through 234580:

```
//stepname EXEC TAPEINFO  
VOL=234560-234580,SHORT
```

Example

To print all the information available for tapes 123456, 234567, and 345678 with each printout starting on a new page:

```
//stepname EXEC TAPEINFO  
VOL=123456  
VOL=234567,NEWPAGE  
VOL=345678,NEWPAGE
```

6.1.2 VOLLIST

VOLLIST finds all tapes owned by a particular userid, all tape data sets owned by a particular userid, or all tape data sets with a particular string in the data set name, even though they may be owned by various userids. It uses the MAX/BATCH program to search the tape library and list information about each data set on tape. The information includes the volume serial number, name of the data set on the tape, file sequence number, creation date, expiration date, date last used, name of the job that created the tape, whether the tape is part of a multi-volume set, and whether the tape will expire within 60 days. The information is current as of shortly after

midnight each day. Changes made to a tape after that time will be shown on the next day.

Although VOLLIST identifies multi-volume tapes (i.e., those whose data spans more than one volume) it does not indicate which tapes make up that group. The TAPEINFO procedure (see section 6.1.1) can be used to list information about all of the tapes that are part of a particular multi-volume group.

The list of tape data sets is sorted by volume serial number. To request that the list be sorted by data set name, use the symbolic parameter SORT.

Optional Symbolic Parameter

SORT Specifies whether list of tapes should be sorted by volume serial number (the default) or by data set name. SORT=BYDSN sorts the list by data set name.

You need at least one control statement to specify the search parameters. The format of the control statement is:

```
LIST IF(column,length,C'searchfield')[ AND(60,EQ,C' 1') ]
```

Required Control Statement and Parameters

LIST IF Identifies the statement as the control statement. To continue the control statement on the next line, place a comma at the end of the first line. The continued part of the control statement may begin in any column.

column A number specifying the first column of the search field. If the search field is part of the data set name, the number should be between 13 and 56.

length A number indicating the length of the field to be searched or EQ to indicate an exact match to the search field.

searchfield The string to search for, usually a userid. The string is case sensitive. Separate multiple strings with commas. Tapes matching any one of the strings are selected.

AND(60,EQ,C' 1') Specifies that only the first data set on each tape is to be listed. There must be at least one blank before the word "AND". If this parameter is omitted, all data sets on each tape are listed.

Example

The following example lists all tapes that have the userid \$iii beginning in column 13. Because the string "AND(60,EQ,C' 1')" is included, only the first data set on each tape is listed. Since tapes are owned by the userid of the first data set on the tape, VOLLIST effectively lists all tapes owned by the userid \$iii. The period in the search string limits the search to a high-level qualifier that exactly matches the search string.

```
//stepname EXEC VOLLIST
LIST IF(13,EQ,C'$iii.') AND(60,EQ,C' 1')
```

Example

The following example lists all tape data sets that have the userid \$iii beginning in column 13. The period in the search string limits the search to a high-level qualifier that exactly matches the search string.

```
//stepname EXEC VOLLIST
LIST IF(13,EQ,C'$iii.')
```

Example

The following example lists all tape data sets that use the North-style naming convention of beginning the data set name with account followed by a period and the initials.

```
//stepname EXEC VOLLIST
LIST IF(13,EQ,C'aaa.iii.')
```

Another way to code this would be to specify that the initials begin in column 17.

```
//stepname EXEC VOLLIST
LIST IF(17,EQ,C'iii.')
```

Example

The following example lists all tapes that have the userid \$iii, aaaaiii, or \$jjj beginning in column 13. Because the string "AND(60,EQ,C' 1')" is included, only the first data set on each tape is listed. Since tapes are owned by the userid of the first data set on the tape, VOLLIST effectively lists all tapes owned by either \$iii, aaaaiii, or \$jjj. The period in the search string limits the search to high-level qualifiers that exactly match the search strings. The SORT=BYDSN parameter specifies that the list be sorted by data set name instead of volume serial number.

```
//stepname EXEC VOLLIST,SORT=BYDSN
LIST IF(13,EQ,C'$iii.,aaaaiii.,$jjj. '),
      AND(60,EQ,C' 1')
```

Example

The following example lists all tape data sets that include the string MASTER anywhere in the data set name regardless of who owns the tape. It searches columns 13 through 56 (44 columns) for the string MASTER. The SORT=BYDSN parameter specifies that the list be sorted by data set name instead of volume serial number.

```
//stepname EXEC VOLLIST,SORT=BYDSN
LIST IF(13,44,C'MASTER')
```

6.1.3 TI

The TI utility displays information about a tape, such as volume serial number, media type, creation date and time, creation program, job name, and last used date and time. The TI utility is part of the CA-1 Tape Management System. To use the TI utility:

1. On the ISPF Data Set List Utility panel (3.4), enter the data set name in the Dsname Level field.
2. On the DSLIST panel, enter TI next to the data set name.

Titan displays tape information on the CA-1 Tape Inquiry panel.

The ISPF Data Set List Utility panel (3.4) displays only cataloged data sets. To display information about uncataloged tape data sets, use the CA-1 Primary Option Menu (ISPF panel L.5, see section 6.1.4), the TAPEINFO batch procedure (see section 6.1.1), or the VOLLIST procedure (see section 6.1.2).

6.1.4 CA-1 Primary Option Menu

Using the CA-1 Primary Option Menu (ISPF panel L.5), you can display information about a tape, such as volume serial number, creation date and time, creation program, job name, and last used date and time. This panel is part of the CA-1 Tape Management System. To access the panel:

1. On the ISPF Primary Option Menu, select L (Local)
2. On the Local Applications and Utilities panel, select 5 (Tape Inventory)
3. On the CA-1 Primary Option Menu, select 1 (INQ/UPD)

The CA-1 Inquiry/Update TMC Menu is displayed.

Selecting tapes

You can select tapes by their volume serial numbers or their data set names.

- The most efficient search results from entering a single volser as the tapes are indexed by volser.

-
- If you select a range of volsers, CA-1 will display ALL tapes in the specified range regardless of the owner of the tape. This can take a significant amount of time if the range is large.
 - You can use a wildcard to select all tapes whose data set names begin with the characters you specify.

CA-1's wildcard is "-" not "*". For example, *iii.-* not *iii.**

Since the panel must search the entire CA-1 data base, this search also takes a significant amount of time. Searching by data set name is not efficient.

1. On the CA-1 Inquiry/Update TMC Menu, enter the volser numbers or data set names for a tape or range of tapes.

The information is displayed on the Volser Inquiry Display panel.

Modifying the Expiration Date or Scratching a Tape

You can modify the expiration date on a tape or scratch a tape data set.

1. On the Volser Inquiry Display panel, enter: U
2. On the Volser Update Display panel, enter a new expiration date in the EXPDT field.
3. To scratch a tape, set the expiration date for the first data set on the tape to today's date.

The tape will become eligible for reuse when the tape management process runs at about 6:00 am the next morning. This will happen even if the tape has multiple data sets. All data sets cataloged on that tape will be uncataloged.

6.2 Mapping Tapes

6.2.1 TAPEMAP

TAPEMAP prints tape labels for foreign tapes, scans for block count, and reports the size of the largest block. It executes the FATS/FATAR program from Innovation Data Processing.

TAPEMAP is useful for determining whether a foreign tape has labels. It also reports the names of the data sets and the tape density.

TAPEMAP prints the labels for tapes with IBM standard labels or ISO/ANSI labels. For these tapes TAPEMAP displays the formats, and layouts of the volume label (VOL1), header labels (HDR1, HDR2), and all trailer label information for one data

set or for all data sets on a tape. If a tape is unlabeled (NL) TAPEMAP treats all files on that tape as data.

Appropriate messages are printed for error conditions such as I/O errors, data checks, incorrect density, tape marks embedded in data, and missing labels. Because of the way it functions, TAPEMAP does not read the volume label to verify that the correct tape has been mounted; so there is a slight chance that the wrong tape may be mapped.

TAPEMAP can be used only with foreign tapes whose volume serial number **does not** match an entry in the Tape Management System. If your tape number matches a number in TMS, your job will ABEND with an S9EC and you will see the message:

```
IEFTMS50 9XX-12 . . .
```

TAPEMAP can handle only one tape volume at a time. It is not designed to be used as a step within a job that makes other references to the tape. It should be used as a single-step diagnostic tool only.

Required Symbolic Parameters

TAPE Supplies the volume serial number of the tape from which the data is being mapped.

Optional Symbolic Parameters

DRIVE Supplies the unit type of the tape device being used. Defaults to DRIVE=FRGN3490.

DSNUM Identifies the data sets to be mapped. The options are:

1 Maps the first data set. This is the default.

ALL Maps all data sets on the tape.

To map the first data set on a foreign tape:

Example

```
//stepname EXEC TAPEMAP,TAPE=tapeser
```

To map and scan all the data sets on a foreign tape:

Example

```
//stepname EXEC TAPEMAP,DSNUM=ALL,TAPE=tapeser
```



7 Transfer Utilities

7.1 Transferring Data Between Workstation and Host

Data transmitted to or from remote workstations by JES2 is either card image or print image. Two procedures, RMTDSIN or RMTDSOUT, are used to transmit tape data containing blocks of different sizes.

The tape transmission schemes supported by the NIH Computer Center have been developed for use with DATA100, Mohawk, and Harris remote workstations. IBM workstations do not support tape transmission. Consult the documentation supplied by the vendor of the particular remote workstation for complete technical information on how to have the workstation initiate transmission of data from the tape drive in the proper format.

Both RMTDSIN and RMTDSOUT return a condition code of zero if there are no errors. Any errors will be listed in the job output and will result in a condition code of four or eight. If the programs invoked by these procedures are unable to write the error message, a condition code of twelve is generated.

These procedures cannot process multiple files in a single execution and are not designed to handle multi-data-set tapes.

The NIH Computer Center does not recommend the use of tape transmissions longer than one hour since it is necessary to restart the transmission from the beginning if the workstation is disconnected for any reason. If it is necessary to disconnect remotes because of problems at the central site, it will be done regardless of tape transmissions in progress at the time.

7.1.1 RMTDSIN

RMTDSIN sends tape data from a remote workstation to the central facility. The program executed by the RMTDSIN procedure reads the data and rewrites it in its original format on a disk or tape. The transmission of data with fixed, variable and undefined length records is supported. With variable length records, RMTDSIN assumes that the correct block and record descriptor words are included in the data transmitted.

Required Symbolic Parameters

NAME Specifies the name of the output data set, which will contain the data transmitted from the remote. A temporary data set would be specified as NAME='&dsname.'

Optional Symbolic Parameters

OPTIONS	Specifies which format of the character count record to use. Converting data to and from card images involves the insertion of a character count record before each group of card images representing a block of data on tape. The character count record can be one of three formats. DATA100 is the default. Code MOHAWK or HARRIS for those types of RJE workstations.
STORAGE	Specifies the unit name for the output data set. The default is LONGTERM.
RECFM	Specifies the record format of the stored data set. Should be the same as the original format; the default is U.
LRECL	Specifies the record length of the stored data set. Should be the same as the original record length; the default is 13030.
BLKSIZE	Specifies the block size of the stored data set. Should be the same as the original block size; the default is 13030.
STATUS	Specifies whether the stored data set is new or old; the default is NEW. Code OLD if the data set already exists.
STEPEND	Indicates a disposition for output the data set; CATLG is the default.
DISK	Specifies the volume serial number of the output disk or tape where the data will be stored. Only required for a data set written to a dedicated disk.

The following four parameters are used only for allocating space for new disk data sets. The default values are sufficient to hold one-to-three million bytes of data on a 3390 disk volume, depending on the block size.

UNITS	Specifies the units of space to be allocated; the default is TRK for tracks. Code CYL for cylinders or the block size for the average block length method.
SIZE	Specifies the number of units in the primary allocation; the default is 10.
INCR	Specifies the number of units in each secondary allocation (up to 15 additional allocations allowed); the default is 5.
UNUSED	Specifies whether any unused tracks should be released or retained; the default is RLSE to release. Code UNUSED= to retain unused space.

The following four parameters are used only when creating a tape data set at the central facility:

- DEN Specifies the density of the tape being created at the NIH Computer Center. DEN should be specified whenever a 9-track tape is accessed.
- SEQ Specifies the data set sequence number of the data set on the tape at the NIH Computer Center. The default for SEQ is 1. Needed only when the data set is not the first one on the tape.
- LBL Specifies the type of labels on the tape at the NIH Computer Center. Needed only when the tape does not contain standard IBM labels. Code LBL=NL if the tape is unlabeled.

The RMTDSIN job is read in from magnetic tape data. Then the operator follows the technique defined by the vendor for initiating transmission of data from the tape.

To send magnetic tape data from a Harris remote workstation to the central facility for storage as a cataloged data set in the LONGTERM management class:

Example

```
// EXEC RMTDSIN,NAME='userid.dsname',  
//      OPTIONS=HARRIS  
//SYSIN DD DATA  
      (data from magnetic tape)
```

7.1.2 RMTDSOUT

RMTDSOUT transfers data from the central facility to the remote site. The program executed by RMTDSOUT generates a data set (SYSOUT Class C) containing the deblocked data, which is routed to the remote tape unit. Variable length records are transmitted with the descriptor words included.

SYSOUT Class C is used for the purpose of transmitting data to a remote tape unit. The remote tape device should be set to accept only SYSOUT=C data sets. For additional information, refer to the *Remote Job Entry Workstation Guide*.

Required Symbolic Parameters

- NAME Specifies the name of the input tape or disk data set to be transmitted to the remote. A temporary data set would be specified as NAME='&dsname'.

Optional Symbolic Parameters

STORAGE	Specifies the unit name for the data set to be transmitted. Required if the data set is not cataloged.
OPTIONS	Specifies which format of the character count record to use. DATA100 is the default. Code MOHAWK or HARRIS for those types of RJE workstations.
RECFM	Specifies the record format of the data set to be transmitted. The default action is to use the RECFM in the data set labels. Therefore, this parameter is needed only for unlabeled tapes, or to override the record format in the data set labels.
LRECL	Specifies the record length of the data set to be transmitted. The default action is to use the LRECL in the data set label. Thus, this parameter is needed only for unlabeled tapes or to override the value in the data set labels.
BLKSIZE	Specifies the block size of the data set to be transmitted to the remote. The default is to use the block size from the data set labels. Thus, this parameter is needed only for unlabeled tapes or to override the value in the data set labels.
STEPEND	Specifies disposition of the data set to be transmitted; the default is KEEP. New data sets on public disks are automatically cataloged.
DISK	Specifies the volume serial number of the tape or disk containing the data set to be transmitted. Required only if the data set is not cataloged.

The following parameters are used only when transmitting from a tape data set at the central facility:

DEN	Specifies the density of the tape being transmitted from the NIH Computer Center. DEN should be specified whenever a 9-track tape is accessed.
SEQ	Specifies the data set sequence number of the data set on the tape at the NIH Computer Center. The default for SEQ is 1. Needed only when the data set is not the first one on the tape.
LBL	Specifies the type of labels on the tape at the NIH Computer Center. Needed only when the tape does not contain standard IBM labels. Code LBL=NL if the tape is unlabeled.

To receive data for magnetic tape storage at a DATA100 remote workstation from a cataloged data set on an online public disk at the central facility:

Example

```
/*ROUTE OUTPUT REMOTEn (if the job is not submitted
    from the remote)
// EXEC RMTDSOUT,NAME='userid.dsname'
```

7.2 Transferring Data from Host to Host

For transferring files between the NIH Computer Center and other computer centers, a host-to-host file transfer can be performed using SENDFILE and RCVFILE. Another available method is CONNET:Direct.

7.2.1 SENDFILE and RCVFILE

The SENDFILE and RCVFILE programs are used with the /*XMIT statement to transfer files between hosts. The following restrictions apply:

- The other site must be defined to NIH as a JES2 NJE node. That is, it must be an OS/390 JES2 or JES3 node, a VM RSCS node, or a VSE POWER node.
- The other site must have the NIH Computer Center-written SENDFILE and RCVFILE programs (or equivalent programs for VM or VSE).
- The user transferring a file must be authorized to use both the sending and receiving site.
- Any job that is submitted to Titan must use the USER and PASSWORD parameters on the JOB statement to specify the Titan userid and password.
- To view the job in OUTPUT HOLD on Titan rather than printing immediately, add a /*ROUTE PRINT NIHJES2.HOLD statement.
- If you run a job at another NJE node and then route the job to print on Titan, you must add a /*JOBPARM ROOM=bbbb statement as described in section 2.3.1.
- The job must contain a /*NETACCT statement as described in section 2.3.1.

The following nodes are defined to Titan:

Node Name	Data Processing Center
NIHJES2	NIH Titan production
NIHJESDR	NIH Titan Disaster Recovery Site
BFD	FDA Center for Food Safety

CDCJES2	Centers for Disease Control & Prevention, Atlanta, GA
SSAPRD1	Social Security Administration, Baltimore, MD
SSAPRD3	Social Security Administration, Baltimore, MD
HCFJES	Centers for Medicare & Medicaid Services (formerly HCFA), Baltimore, MD
CCF1	BLS, SunGard, Voorhees, NJ
NPH	BLS, SunGard, Voorhees, NJ
SPGFLD	Dept. of Commerce, Springfield, VA

These nodes have the SENDFILE and RCVFILE programs. Other defined nodes may obtain them upon written request from the NIH Computer Center. Non-OS/390 nodes should modify the programs to suit their environment.

SENDFILE reads the data set to be transferred and converts it into 80-byte records for transmission, adding the necessary control information to permit the data set to be reconstructed at the receiving site.

RCVFILE reconstructs a data set from the 80-byte records created by SENDFILE. The RCVFILE program is executed at the receiving site.

To transfer nonsequential data sets, first create a sequential, transportable copy (e.g., an IEBCOPY unload of a PDS or an AMS EXPORT or REPRO of a VSAM data set). You can rebuild the nonsequential datasets at the receiving node.

SENDFILE DD Statements

SYSUT1	Defines the data set to be sent. Variable-length spanned records cannot exceed 32760 bytes.
SYSUT2	Specifies the output containing the job stream and converted data set for the receiving end. RECFM=FB,LRECL=80,BLKSIZE=80 (default) or multiple of 80. Normally defined as SYSOUT=(A,INTRDR).
JCL	Specifies a job stream to be executed at the receiving host to reconstruct data set. This jobstream must always include a job step that executes the RCVFILE program.

RCVFILE DD Statements

- SYSPRINT** Specifies the place where messages are written. Normally a SYSOUT data set. RECFM=FBA,LRECL=121,BLKSIZE=121 (default) or multiple of 121.
- SYSUT1** For a sequential data set, normally specified as * and appears immediately before the delimiter.
- For a PDS or a job requiring further processing, the special control statement ****SENDFILE**** appears immediately after the SYSUT1 DD statement, and must be in columns 1/12. The ****SENDFILE**** record signals SENDFILE to insert the transferred data set at this point. This is necessary when another job step follows RCVFILE.
- SYSUT2** Defines the data set at the receiving node. The RECFM, LRECL, and, if not overridden, BLKSIZE will be those of the original data set. If a new BLKSIZE is specified, the following should be noted:
- For fixed-length records, if BLKSIZE is not equal to LRECL, the "blocked" indicator will be set.
 - For variable-length records, if BLKSIZE is less than LRECL+4, the "spanned" indicator will be set; otherwise, the "spanned" indicator will be turned off.
 - For undefined-length records, the new BLKSIZE must be at least as large as the original BLKSIZE; otherwise, a S013-34 ABEND will occur.

Example

This example shows the transfer of a sequential data set. The sending host is Titan; the receiving host is identified as "nodename." The JCL between the Titan JOB statement and the /*XMIT statement must conform to the standards of the sending site (Titan). The JCL after the receiving node JOB statement must conform to the standards of the receiving site.

The delimiter definition (DLM=) must identify a unique character sequence as the delimiter. In the example, the delimiter is XX.

```
//IIISEND1 JOB (Titan JOB statement)
//SEND EXEC PGM=SENDFILE
//SYSUT1 DD DSN=userid.DATASET,DISP=SHR
//SYSUT2 DD SYSOUT=(A,INTRDR)
//JCL DD DATA,DLM=XX
//IIIXMIT JOB (Titan JOB statement)
```

```

/*NETACCT titanuserid
/*XMIT nodename
//rcvjob JOB (receiving node JOB statement)
/*JOBPARM ROOM=bbbb
/*ROUTE PRINT NIHJES2.HOLD
(any other control statements required by receiving host)
//RECEIVE EXEC PGM=RCVFILE
//SYSPRINT DD SYSOUT=*
//SYSUT2 DD (parameters defining reconstructed data set)
//SYSUT1 DD *
XX

```

In the SEND step, the SYSUT1 DD statement specifies the sequential data set to be transferred. The SYSUT2 DD statement specifies the output containing the job stream and converted data set. The JCL DD statement specifies a job stream to be executed at the receiving host to reconstruct data set userid.DATASET at receiving site “nodename.” This jobstream must include a job step that executes the RCVFILE program.

In the RECEIVE step, the SYSPRINT DD statement specifies a message data set. The SYSUT2 DD statement specifies the data set that was transmitted and is to be reconstructed at the receiving site. The SYSUT1 DD statement must be the last JCL statement prior to the delimiter XX.

Example

This example shows the transfer of a PDS. The sending host is Titan; the receiving host is identified as “nodename.” The JCL between the Titan JOB statement and the /*XMIT statement must conform to the standards of the sending site (Titan). The JCL after the receiving node JOB statement must conform to the standards of the receiving site.

The delimiter definition (DLM=) must identify a unique character sequence as the delimiter. In the example, the delimiter is XX.

```

//IIISEND2 JOB (Titan JOB statement)
//UNLOAD EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=A
//IN DD DSN=userid.PDS,DISP=SHR
//OUT DD DSN=&&TEMP,DISP=(,PASS),UNIT=SYSDA,
// SPACE=(CYL,(1,1),RLSE)
//SYSIN DD *
COPY INDD=IN,OUTDD=OUT
/*
//SEND EXEC PGM=SENDFILE,COND=(0,NE)
//SYSUT1 DD DSN=&&TEMP,DISP=(OLD,DELETE)
//SYSUT2 DD SYSOUT=(A,INTRDR)
//JCL DD DATA,DLM=XX
//IIIXMIT JOB (Titan JOB statement)

```

```

/*NETACCT titanuserid
/*XMIT nodename DLM=YY
//rcvjob JOB (receiving node JOB statement)
/*JOBPARM ROOM=bbbb
/*ROUTE PRINT NIHJES2.HOLD
(any other control statements required by receiving host)
//RECEIVE EXEC PGM=RCVFILE
//SYSPRINT DD SYSOUT=*
//SYSUT2 DD DSN=&&TEMP,DISP=(,PASS),UNIT=SYSDA,
// SPACE=(CYL,(1,1),RLSE)
//SYSUT1 DD *
**SENDFILE**
/*
//LOAD EXEC PGM=IEBCOPY,COND=(0,NE)
//SYSPRINT DD SYSOUT=*
//IN DD DSN=&&TEMP,DISP=(OLD,DELETE)
//OUT DD (parameters defining reconstructed data set)
//SYSIN DD *
COPY INDD=IN,OUTDD=OUT
/*
YY
XX

```

The first step of this job unloads the PDS named `userid.PDS` to a temporary sequential file named `&&TEMP`, which is then passed to the next job step (named `SEND`) for transmission to the receiving host named “nodename.” The `SENDFILE` step will not be executed if something goes wrong with `IEBCOPY` (controlled by `COND=(0,NE)` on the `EXEC` statement).

The `/*XMIT` statement must contain a delimiter (`YY`) because the data stream that follows `/*XMIT` contains statements beginning with `/*`.

The `**SENDFILE**` statement is included immediately after the `SYSUT1 DD` statement. `**SENDFILE**` signals `SENDFILE` to insert the transferred data set (`&&TEMP`) at this point. The `LOAD` step follows `RCVFILE`. This last job step executes `IEBCOPY` to reconstruct the transmitted sequential data set named `&&TEMP` into a PDS.

SENDFILE and RCVFILE Completion Codes

`SENDFILE` returns a completion code of 0 if `SYSUT1` does not define a null data set, and 4 if it does. `SENDFILE` issues a U0100 ABEND if a `DD` statement is omitted.

`RCVFILE` returns a completion code of 0 if the data set is successfully reconstructed, 4 if `SYSUT1` specifies a null data set, and 12 if an error occurs (forcing `RCVFILE` to terminate). The `SYSPRINT` message indicates the execution result. `RCVFILE` issues a U0100 ABEND if a `DD` statement is omitted.

7.2.2 CONNECT:Direct

CONNECT:Direct is a commercial product that provides host-to-host file transfer. The function it provides is similar to that provided by the SENDFILE and RCVFILE programs discussed above. CONNECT:Direct is easier to use than those two programs and provides a means to monitor the progress of the file transfer.

There are two requirements for use of CONNECT:Direct: there must be a VTAM or TCP/IP connection between NIH and the other site, and the other site must also have the hub version of CONNECT:Direct.

The actions carried out by CONNECT:Direct are controlled by a CONNECT:Direct process, which is similar to a JCL procedure. Once the process is set up, the user needs to supply just the values for various parameters. The two parameters used most frequently are the source data set name and the target data set name.

The NIH Computer Center's CONNECT:Direct node name, APPLid, and NETid are:

System	Node Name	APPLid	NETid
Titan	nih.std.ndm	NIH4NDM	NIH

Anyone wishing to set up a new CONNECT:Direct application must be a registered CIT user. Contact the NIH Help Desk or the account sponsor for your organization if you do not already have a CIT userid. Once you are a registered CIT user, you can then register for using CONNECT:Direct. Use of CONNECT:Direct requires coordination with another site as well as modifications to certain CONNECT:Direct configuration files.

To register to use CONNECT:Direct, submit a Service Request (see section 1.2) that includes the following information:

- destination of the data to be transmitted
- nature of the data to be transmitted
- the amount and frequency of transmitted data

Each project must have a Data Transmission Administrator (DTA), who is responsible for setting up and maintaining the jobs and CONNECT:Direct processes needed to transmit the data. All correspondence with CIT regarding a registered CONNECT:Direct project should specify the name of the DTA.

If users on other systems will be transferring files to Titan via CONNECT:Direct, the DTA must obtain a restrictive Titan userid for each such user. This userid will have the following special characteristics:

- logon to NIH systems will not be permitted
- the associated password will not expire

Users from other systems must add the following CONNECT:Direct parameter to the appropriate CONNECT:Direct control statements:

```
SNODEID=(userid,password)
```

The DTA must arrange that an appropriate RACF profile be set up and that this restricted userid has access to it. The DTA should contact the security coordinator for the account.

Data Transmission Administrators should refer to the following manuals:

CONNECT:Direct for MVS User's Guide

This manual introduces the command system through which the CONNECT:Direct Processes are controlled. It describes the commands available and tells how to use them through the batch interface. The manual assumes knowledge of the IBM Multiple Virtual Service operating system and job control language.

CONNECT:Direct Process Guide

This manual provides information needed to write CONNECT:Direct Processes. A CONNECT:Direct Process is a unit of work. Processes are written in a language unique to the CONNECT:Direct product. The CONNECT:Direct Process language consists of statements and parameters that provide instructions for initiating such activities as copying files from one CONNECT:Direct node to another.

The two manuals listed above can be obtained from the vendor, Sterling Commerce (<http://www.sterlingcommerce.com/>).

CONNECT:Direct is available on Titan through the batch interface and the interactive user interface. TCP/IP and SNA are the only network connectivity protocols supported for CONNECT:Direct at CIT.

DMBATCH

DMBATCH is a batch interface program for CONNECT:Direct. The following example shows the JCL for using DMBATCH. Change the control statements to reflect your request requirements.

Example

```
//jobname JOB (Titan JOB statement)
//*+JBS BIND CONNECT
//DMBATCH EXEC PGM=DMBATCH,REGION=1024K,PARM=(YYSLYNN)
//DMNETMAP DD DISP=SHR,DSNAME=NIH.NDM.NETMAP
//DMPUBLIB DD DISP=SHR,DSNAME=NIH.NDM.PROCESS
//DMMSGFIL DD DISP=SHR,DSNAME=NIH.NDM.MSG
```



```
//DMPRINT DD SYSOUT=*  
//SYSIN DD *
```

8 Job Utilities

8.1 Checking for JCL Errors

8.1.1 JCLSCAN

JCLSCAN scans the user's JCL for syntactical errors without actually scheduling any part of the job. Therefore no tapes or disks will be requested, and no programs will be executed.

A common reason for job failure is JCL errors. JCLSCAN is more effective and less costly than testing JCL by executing a run. It is not a special program; it simply uses the standard OS/390 Reader/Interpreter.

You can place the JCLSCAN step at any point in your job. The logical place is either before the first EXEC statement or after the last one. If your job ends with null statement (//) the JCLSCAN step must appear before the null statement.

Example

```
//jobname JOB (Titan JOB statement)
      .
      .
      (user JCL)
      .
      .
//stepname EXEC JCLSCAN
```

Example

In this example, the job ends with a null statement.

```
//jobname JOB (Titan JOB statement)
      .
      .
      (user JCL)
      .
      .
//stepname EXEC JCLSCAN
//
```

Example

```
//jobname JOB (Titan JOB statement)
//stepname EXEC JCLSCAN
//stepname EXEC (first step of original job)
.
.
(more user JCL)
.
.
```

An in-stream procedure will not be scanned unless there is an EXEC statement for it.

The messages from this procedure are self-explanatory. The first error message, IEF645I, is caused by a deliberate error (an invalid referback) in the procedure that forces the rest of the user's JCL to be scanned. All other error messages pertain to the user's JCL.

If the scan completes, the output will show the error message IEF645I and any other error messages caused by the JCL. If the error message IEF645I does not appear in the output, the scan was stopped by one or more syntax errors. When these have been corrected, JCLSCAN should be run again to check for additional errors.

8.2 Submitting Batch Jobs

This section discusses the DSSUBMIT procedure and the DSSUBMIT program.

8.2.1 DSSUBMIT Procedure

The DSSUBMIT procedure reads a data set from disk and submits the data set as a batch job. This procedure uses the IBM utility program IEBGENER to read the data set and pass it directly to the JES2 internal reader (INTRDR), which reads and interprets the data set. The format of the data set may be fixed or fixed block with an LRECL up to 254. DSSUBMIT is an NIH-written utility.

If a job should not be run unless a previous job terminated successfully, make the DSSUBMIT procedure the last step of the previous job. Through condition code testing, you can ensure that the DSSUBMIT step does not submit the job if previous steps have failed.

Because using the DSSUBMIT procedure permits precise control of job executions for a series of jobs, it can be used to avoid data set contention problems (see section 2.3.1, "/*BEFORE and /*AFTER").

Each data set submitted must contain all necessary JCL. These statements must be standard JCL and cannot exceed 80 characters in length. However, the input stream data may be up to 254 characters long.

Required Symbolic Parameters

NAME Names the data set containing the job to be submitted.

Optional Symbolic Parameters

DISK Specifies the volume serial number of the disk on which the data set is stored. Required if the data set is not cataloged.

STORAGE Gives the unit name for the input device if it is not cataloged.

STEPEND Indicates the final (end of step) disposition for the input data set; KEEP is the default.

Example

```
//stepname EXEC DSSUBMIT,NAME='userid.dsname'
```

It is not necessary that the job to be submitted be saved in a data set. The following example illustrates how the job may be included in the input stream.

Example

```
//stepname EXEC DSSUBMIT
//JOB DD DATA,DLM='><'
//JDASUBT2 JOB (),'SUBMITTED JOB 2',MSGCLASS=H
//*
//JUNK EXEC PGM=IEFBR14
><
```

8.2.2 DSSUBMIT Program

The DSSUBMIT program is an NIH-written utility that uses an old, unsupported interface to submit jobs to JES2. While it still works in most situations, it is recommended that you modify your JCL to pass the job to the internal reader (INTRDR) directly. A procedure named DSSUBMIT is available for this purpose (see section 8.2.1).

8.3 Fetching Jobs

8.3.1 FETCH

FETCH copies job output to a data set. If the job has not finished execution, FETCH retrieves only the output that is available. You may wish to include the `/*AFTER` statement to ensure that the job finishes execution before the output is fetched. The format is:

```
//stepname EXEC FETCH,PARM='parameter=option,parameter=option...'
//SYSUT2 DD DSN=userid.dsname,DISP=(disp,CATLG),UNIT=unit,
//   DCB=(LRECL=133,BLKSIZE=0,RECFM=FB),
//   SPACE=(space,(nn,nn),RLSE)
```

Required parameter. You must specify either the job name or job number.

J The job name

JI The letter J followed by the job number

Optional parameters. You can specify any of the following parameters to further identify a job or group of jobs or to determine the disposition.

Q The sysout class

STATUS The output status. HELD selects output that is held, NONHELD selects output that is not held, and ALL selects output that is held or not held (all output). The default is ALL.

D The destination for the job output

F The form specified for the job

W The writename, the same as the SYSOUT subparameter *program*. Not used.

NEWDISP Either KEEP or PURGE the output after it has been fetched. The default is PURGE.

Example

This JCL will fetch the job and purge the job output.

```
//stepname EXEC  FETCH,PARM='JI=J04080'
//SYSUT2 DD  DSN=userid.dsname,DISP=(NEW,CATLG),UNIT=DISK2YR,
//  DCB=(LRECL=133,BLKSIZE=0,RECFM=FB),
//  SPACE=(TRK,(10,1),RLSE)
```

Example

This JCL will fetch the job and KEEP the job output.

```
//stepname EXEC  FETCH,PARM='JI=J04081,NEWDISP=KEEP'
//SYSUT2 DD  DSN=userid.dsname,DISP=(NEW,CATLG),UNIT=DISK2YR,
//  DCB=(LRECL=133,BLKSIZE=0,RECFM=FB),
```

```
// SPACE=(CYL,(10,10),RLSE)
```

Example

Two jobs named STICK are in the output queue. One is sysout class H and one is sysout class A. Both will be fetched.

```
//stepname EXEC  FETCH,PARM='J=STICK,NEWDISP=KEEP'  
//SYSUT2 DD  DSN=userid.dsname,DISP=(NEW,CATLG),UNIT=DISK2YR,  
//  DCB=(LRECL=133,BLKSIZE=0,RECFM=FB),  
//  SPACE=(CYL,(10,10),RLSE)
```

Example

Two jobs named STICK (4081,4082) are in the output queue. Job 4081 is sysout class H (held), job 4082 is sysout class A (not held). Job 4081 will be fetched.

```
//stepname EXEC  FETCH,PARM='J=STICK,NEWDISP=KEEP,STATUS=HELD'  
//SYSUT2 DD  DSN=userid.dsname,DISP=(NEW,CATLG),UNIT=DISK2YR,  
//  DCB=(LRECL=133,BLKSIZE=0,RECFM=FB),  
//  SPACE=(CYL,(10,10),RLSE)
```

Example

Two jobs named STICK (4081,4082) are in the output queue. Job 4081 is sysout class H, job 4082 is sysout class A. Job 4082 will be fetched.

```
//stepname EXEC  FETCH,PARM='J=STICK,NEWDISP=KEEP,Q=A'  
//SYSUT2 DD  DSN=userid.dsname,DISP=(NEW,CATLG),UNIT=DISK2YR,  
//  DCB=(LRECL=133,BLKSIZE=0,RECFM=FB),  
//  SPACE=(CYL,(10,10),RLSE)
```

Example

A job named STICK is in the output queue and has msgclass H (held) and sysout class A (not held). The JES2 job log will be fetched.

```
//stepname EXEC  FETCH,PARM='J=STICK,NEWDISP=KEEP,STATUS=HELD'  
//SYSUT2 DD  DSN=userid.dsname,DISP=(NEW,CATLG),UNIT=DISK2YR,  
//  DCB=(LRECL=133,BLKSIZE=0,RECFM=FB),  
//  SPACE=(CYL,(10,10),RLSE)
```

9 Language Utilities

9.1 Language Procedures

The following sections describe the language compilers supported on Titan. For additional information, consult the appropriate documentation, available from:

<http://publications.cit.nih.gov>

9.1.1 FORTRAN

A FORTRAN compiler is available interactively or in batch mode from TSO/ISPF. The foreground interactive compiler is on ISPF panel 4.3; the background batch compiler is on ISPF panel 5.3.

The following table lists the cataloged procedures for batch processing:

FORTRAN Cataloged Procedures

Procedure Name	Key Letters	Function Performed
FORTC	C	Compiles a source program
FORTCL	CL	Compile and link edit
FORTCLG	CLG	Compile, link edit, and execute
FORTCG	CG	Compile and execute

Note: The GO step in each of the execute procedures provides the DD statements, which assume data set reference number 06 for printed output.

For precise information about using each procedure, see the compiling section in the IBM FORTRAN programming guide available on the CIT Publications web page (see section 1.3). The IBM procedures are named differently from the NIH procedures. Look for the key letters, which indicate the function of the procedure.

9.1.2 COBOL

Common Business Oriented Language (COBOL) is used for non-scientific applications. The version of COBOL in use on Titan at the NIH Computer Center is compatible with the American National Standards Institute (ANSI) standard and contains a number of IBM extensions. COBOL is based on a well-defined, restricted form of English.

A COBOL compiler is available interactively or in batch mode from TSO/ISPF. The TSO/ISPF foreground interactive compiler is on ISPF panel 4.2; the background batch compiler is on ISPF panel 5.2.

The following table lists the cataloged procedures for batch processing.

COBOL Cataloged Procedures

Procedure Name	Key Letters	Function Performed
COBOLC	C	Compile only
COBOLCL	CL	Compile and link edit
COBOLCLG	CLG	Compile, link edit, and execute
COBOLCG	CG	Compile and execute

For precise information about using each procedure, see the compiling section in the IBM COBOL programming guide available on the CIT Publications web page (see section 1.3). The IBM procedures are named differently from the NIH procedures. Look for the key letters, which indicate the function of the procedure.

9.1.3 PL/I

PL/I is a multi-purpose language used in business and scientific applications. PL/I contains most of the capabilities of FORTRAN and COBOL as well as some additional features.

A PL/I compiler is available interactively or in batch mode from TSO/ISPF. The TSO/ISPF foreground interactive compiler is on ISPF panel 4.5; the background batch compiler is on ISPF panel 5.5.

The following table lists the cataloged procedures for batch processing.

PL/I Cataloged Procedures

Procedure Name	Key Letters	Function Performed
PLIC	C	Compile only
PLICL	CL	Compile and link edit
PLICLG	CLG	Compile, link edit, and execute
PLICG	CG	Compile and execute

For precise information about using each procedure, see the compiling section in the IBM PL/I programming guide available on the CIT Publications web page (see section 1.3). The IBM procedures are named differently from the NIH procedures. Look for the key letters, which indicate the function of the procedure.

9.2 Using the Binder

The IBM Program Manager Binder is a service program used to prepare executable programs from object modules. There are three steps in the process of creating an executable program. The first step is to write a program in a source language such as COBOL, FORTRAN, or PL/I. Next, the source program is translated by a compiler into an object module. Finally, the object module is fully resolved by the Binder to load modules or other object modules to create an executable load module.

Load modules can be either fully or partially resolved. Fully resolved means that all the programs necessary to execute are stored in the load module. Partially resolved means that some of the programs necessary to execute (such as run time support routines) are missing. Usually, partially resolved load modules are created by binding with the NCAL option. NCAL can save the cost of storing programs that are already stored in other libraries.

Binder control statements may aid program modification by replacing, deleting, and rearranging control sections. This means that an entire program need not be recompiled when there are changes to only a few of its control sections.

The Binder also reserves storage for the common control sections generated by the Assembler and FORTRAN languages and the static external areas that are generated by PL/I.

For additional information on the Binder and its functions, consult the manual, *IBM DFSMS/MVS Program Management*, SC26-4916.

9.2.1 Estimating Region Size

In OS/390, storage is allocated separately for each job step. Therefore, each step can be restricted to the amount of core it needs.

The NIH Computer Center staff does careful research to select optimum region sizes for its cataloged procedures, and they should generally not be overridden without cause.

Compile and bind steps for each language have what is generally sufficient storage assigned to them by the cataloged procedure. However, the storage required by the GO step is highly dependent on the program being executed. The default value of 4096K provided by the procedure may be too small, causing the job to ABEND. Because of this the user may want to determine the amount of storage a program needs before allowing it to run. This requires submitting a job to obtain a map of the fully resolved load module through use of the Binder. The job consists of a normal compile (e.g., FORTCLG for FORTRAN), with COND.GO=(0,LE) on the EXEC statement.

Example

```
//stepname EXEC FORTCLG,COND.GO=(0,LE)
//FORT.SYSIN DD *
    (source program)
```

Include any Binder control statements needed for actual execution. Since the program will not be executed, there is no need to include any DD statements for the GO step.

The region size in bytes can be determined after this job is run by adding:

total length of program converted to decimal	(given in hexadecimal at the end of the module map.)
+n times buffer size of all data sets which can be opened at any one time	(buffer size = maximum block size.) The default for "n" is 5 for QSAM; 2 for all other access methods. Consult the applicable programmer's guide to determine which access methods are used by the job.
+20,000 bytes	(to cover supervisory services, control blocks, access method routines.)
+ enough to round this to the next higher even multiple of 1024 bytes.	

This figure, divided by 1024 (the value of K), can be entered on the execute statement.

```
REGION .GO=xxxxK
```

9.2.2 Using the Loader

The Loader combines the bind and GO steps into one step. It will accept object modules from compilers or load modules produced by the Binder as input. It will also search libraries defined by the SYSLIB DD statement if unresolved external references remain after processing the primary input from SYSLIN. (To use the automatic search, programs must be stored under the same names by which they are called.) Binder control statements (INCLUDE, ENTRY, OVERLAY, etc.) cannot be used with the Loader.

Using the Loader is often more economical than using the Binder. The Loader does not create an intermediate data set; instead, the program is loaded directly into core and is executed from there. The Loader is best adapted to debugging runs where a different version of the program is compiled and executed each time. If the same program is to be executed many times, it is probably more economical to execute a fully-resolved load module (created with the ILLV LKSM or ILLV LKMM procedures) rather than using either the Binder or the Loader each time.

The entry point for the main program must be specified. Since the entry point is always PLISTART for PL/I, this name is in the PL/I procedures and does not have to be given. For FORTRAN the entry point is always MAIN unless the FORTRAN PROGRAM statement is used, in which case EPT=entry-point must be coded on the EXEC statement invoking the Loader procedure. For COBOL, the PROGRAM- ID is the default entry point unless a specific entry point is coded within the program. For

Assembler, the entry point is coded in the program. To give the entry point, code EPT=entry point on the EXEC statement invoking the Loader procedure.

10 Other Utilities

10.1 Executing TSO Commands at LOGON

To execute a command when you log onto Titan, you can use any or all of the following methods: the TSO/E LOGON panel, the TSLOGON member of your ISPF profile, or the TSLIST member of your ISPF profile. The ISPF profile is named *userid.ISPF.ISPPROF*.

10.1.1 Entering a Command on the TSO/E LOGON Panel

You may enter a single TSO command in the Command field of the TSO/E LOGON panel. For example, if you enter UNITNAME TEMP, your default unit name will be TEMP when you log on (see the UNIT parameter of the DD statement in section 2.6.4).

If you want to execute several commands, save them in a CLIST or a REXX exec and execute that data set. For example, you can enter the command EX '*userid.MY.CLIST*' in the Command field of the TSO/E LOGON panel.

You can also emulate the logon procedure that was used on the South system in which TSO checked for a data set named *aaaaiii.@TSO.PROFILE.CLIST* and, if it found one, executed it. To emulate the procedure, enter EX '*aaaaiii.@TSO.PROFILE.CLIST*' in the Command field of the Titan TSO/E LOGON panel. You will want to make sure that each command in your CLIST is valid for Titan.

After you enter a command in the Command field, it stays there for each logon session until you change it or erase it.

10.1.2 Saving Commands in TSLOGON

When you log onto Titan, the logon procedure checks your ISPF profile (*userid.ISPF.ISPPROF*) for a member named TSLOGON. If the member exists, it is executed as a CLIST. If the member ends with the command SPF, TSO displays the CIT/Titan Primary Option Menu. To go to a particular ISPF panel, specify the option number on the SPF command; for example, specify SPF OPT(2 to go to the Edit Entry Panel. If you do not include an SPF command, TSO displays the READY prompt.

10.1.3 Saving Commands in TSLIST

When TSO searches for commands, you may want it to search your own libraries in addition to the system libraries. To enable this feature, create the TSLIST member in your ISPF profile (*userid.ISPF.ISPPROF*) and add statements containing the following keywords:

Statement keyword	Keyword type	Type of data set identified by keyword	Maximum number of data sets allowed
SYSEXEC	DDNAME	REXX library	7
SYSPROC	DDNAME	REXX/CLIST library	7

The following example shows a simple TSLIST member:

```
SYSEXEC userid.REXX.EXEC
SYSEXEC userid.ISPF.EXEC
SYSPROC userid.CLIST
```

When you log onto Titan, the logon procedure checks your ISPF profile for the TSLIST member. If the member exists, it allocates the data sets to the specified DDNAMEs.

You can also use TSLIST to customize your ISPF panels. Previously, users did this by copying the CIT/Titan Primary Option Menu and adding a link to their own panels. Unfortunately, those users would not see upgrades or new products on the production panel. To solve this problem, the CIT/Titan Primary Option Menu created the USER option for displaying customized panels.

To create a customized panel, you must add statements for the APPLICATION and ISPLLIB keywords to TSLIST. You can also add any of the following keywords:

Statement keyword	Keyword type	Type of data set identified by keyword	Maximum number of data sets allowed
APPLICATION	REXX EXEC or CLIST identifier	REXX library(member) or CLIST library(member)	
ISPLLIB	DDNAME	ISPF link library	3
ISPMLIB	DDNAME	Message library	3
ISPLLIB	DDNAME	Panel library	3
ISPSLIB	DDNAME	Skeleton library	3
ISPTLIB	DDNAME	Table input library	3
ISPTABL	DDNAME	Table output library	3
SYSEXEC	DDNAME	REXX library	7
SYSPROC	DDNAME	REXX/CLIST library	7

The following example shows a simple TSLIST member that will be able to display a customized panel:

```
ISPLLIB userid.ISPF.PANELS
APPLICATION userid.ISPF.EXEC(MYPANELS)
```

When you select the USER option from the CIT/Titan Primary Option Menu, ISPF executes the REXX exec identified in the APPLICATION statement. In the example above, it executes *userid.ISPF.EXEC(MYPANELS)*.

MYPANELS contains the following lines:

```
/* REXX */  
PARSE ARG PARM .  
'ISPEXEC SELECT PANEL(MYAPP) OPT('PARM)'
```

When MYPANELS is executed, the panel *userid.ISPF.PANELS(MYAPP)* is displayed.

For more information about creating customized panels, see the *ISPF Dialog Developer's Guide and Reference (SC28-1273-04)*.

The following rules apply to statements in TSLIST:

- Each statement must start with a keyword.
- At least one space must follow each keyword.
- Data set names must be fully qualified and should not be enclosed in quotes.
- The APPLICATION keyword must reference one and only one REXX exec or CLIST.
- Multiple data sets can be allocated to the same DDNAME. Each data set must be allocated on a separate line. You must also abide by the limits for each DDNAME (see the chart above). If you need to have a limit increased, call the NIH Help Desk at 301-594-6248.

If you select the USER option and you do not have a TSLIST member in your ISPF profile data set, TSO displays the NIH User Application Default Panel, which gives instructions for creating a TSLIST member.

10.1.4 Moving a TSO Session

Occasionally you may need to disconnect a TSO session because you are logged on at another location and need to log on at your current location or because the network line has dropped. You can reconnect to your previous session using the Reconnect option.

The Reconnect option is one of the four options at the bottom of the TSO/E Logon screen on MVS. To use this option, enter an S before the option and log on normally.

You will be disconnected from your previous session and reconnected at your new location.

This method is ideal if you simply want to move your session to another computer. However, if your session is hung, Reconnect will not change the hung state. In that

case, you will need to cancel your session using the Titan SILK Web (see Section 10.1.5).

10.1.5 Canceling a TSO Session

You may need to cancel a TSO session because it is hung or frozen. You can cancel your previous session by using the TSO Command page on the Titan SILK Web. Follow these steps:

1. Go to the Titan Silk Web site at <http://silk.nih.gov/>.
2. Click TSO Commands.
3. List your current jobs
 - Select Status in the Command list box.
 - Enter your Titan userid in the Job or Data Set Name entry box.
 - Press Enter.
4. Respond to the prompt for your Titan userid and password.

Your current jobs are displayed. The one that is executing is your TSO session.

5. Note the job number of your TSO session or highlight and copy it.
6. Return to the TSO Commands page (click the Back button).
7. Cancel your session.
 - Select Cancel in the Command list box.
 - Enter or paste your Titan userid and job number in the Job or Data Set name entry box using the format:
userid(jobnumber)
8. Return to the TSO/E logon screen on MVS to log onto Titan.

10.2 Executing TSO Commands in Batch

10.2.1 BATCHTSO

BATCHTSO enables users to enter TSO commands in a batch job. The formats for the TSO commands is the same in the batch job as it is in an interactive TSO session. The TSO convention governing the use of single quotes around data set names applies. That is, a data set name enclosed in single quotes will be processed as entered, but a data set name not enclosed in single quotes will have the userid under which the job is run added as a prefix automatically before processing.

The following example uses BATCHTSO:

```
//STEPNAME EXEC BATCHTSO
//SYSIN      DD *
.
.
      (TSO commands)
.
.
/*
```

Any number of TSO commands may follow the SYSIN DD statement.

10.3 Executing WYLBUR Commands

This section describes the IBM Session Manager facility, which enables screen scrolling, and the WYLBUR and WYLBURL procedures.

Note: A modified version of NIH WYLBUR is on Titan.

To learn more about the differences between NIH WYLBUR on Titan and NIH WYLBUR on the former South system, go to:

<http://silk.nih.gov/public/PUBLIC.@WWW.TITAN.SOUTH.WYLBUR.HTML>

10.3.1 Scrolling the Screen in WYLBUR

In WYLBUR you can scroll back to output that has disappeared off the screen by using the IBM Session Manager. Without Session Manager, the Titan full screen mode would prevent scrolling back to previous output.

To use Session Manager:

1. On the TSO/E LOGON screen, enter **sessmgr** as the value for Procedure.
2. On the CIT/Titan Primary Option Menu, enter **x**.
You must exit ISPF to use Session Manager. It will not work under ISPF.
3. On the TSO Session Output screen, enter **wylbur**.
You may begin using WYLBUR.
4. To return to ISPF after you have ended your WYLBUR session, enter **SPFMAIN**.

Follow these guidelines:

- Use F7 to scroll backwards and F8 to scroll forwards.

-
- You must define a set break character to be able to leave collect mode. For example:

```
set break BRK
```

You can then exit collect mode by entering BRK as the first characters on any line.

- The modify and retry commands are tricky to use because the cursor is not positioned properly.
- You may sometimes have to enter an extra carriage return to see your output.

For more information about Session Manager, see the *TSO/E User's Guide* on the CIT Publications page at <http://publications.cit.nih.gov/>. Select Time Sharing Option (TSO) under OS/390 Systems.

10.3.2 WYLBUR

The WYLBUR procedure executes WYLBUR commands as a batch job. This makes it possible to run lengthy command procedures, run command procedures that need to be executed at night, or deal with extremely large data sets. Executing WYLBUR commands as a batch job, rather than online, has several advantages:

- It frees up the terminal while WYLBUR commands for lengthy command procedures or large documents are executing.
- A large job can be run at a lower cost than if it were run interactively, especially if the batch job is submitted with the DISCOUNT option for off-hour processing.
- Up to 20 full-size active files can be used. Working interactively, the user is limited to just one full-size active file plus up to 19 smaller active files; the size of the remaining active files is more limited.

The input consists of the WYLBUR commands to be executed. To enter a BREAK, use the SET BREAK command (see the *WYLBUR General Editing* manual) to specify characters to be interpreted as a break signal. A break signal can only be entered as the first response to a prompt; it cannot be used to interrupt a listing, to abort the execution of a command or to ignore information typed.

Input to batch WYLBUR is from a DD statement with the name of SYSIN. Input has the following characteristics:

- It can be in either fixed (RECFM=F) or variable (RECFM=V) length records.
- There is no restriction to card format.

- It is not restricted to 80 columns.
- Input data sets cannot be in edit format.

The output looks like a complete WYLBUR session. The account and registered initials used for the "session" are those specified in the JOB statement.

There are two procedures: WYLBUR and WYLBURL. The only difference between the two is that WYLBURL permits the use of up to 50 active files with each active file using the maximum amount of space. Normally, the WYLBUR procedure should be used; use WYLBURL when several large active files are needed.

There is one optional symbolic JCL parameter available for the WYLBUR and WYLBURL procedures: OPTIONS, which indicates options to be used by the WYLBUR program. The following options are available:

OPTION	PURPOSE
ARGUMENT=	<p>Passes a parameter to be made available to the WYLBUR commands being executed. The word ARGUMENT is followed by an equal sign and a string (a series of characters enclosed in quotes) containing the parameter. The parameter may also be enclosed in parentheses instead of quotes. The parameter is available as the value of the ARGUMENT function at command procedure level 0. (See the <i>WYLBUR Command Procedures</i> manual for more information on the ARGUMENT function and level 0).</p> <p>The ARGUMENT parameter can be made available to command procedures by passing it using the WITH option of EXECUTE, or by storing it in an active file or global variable.</p>
NO ABORT	Causes WYLBUR to continue execution of commands following an error. Normally, if an error occurs, the remaining WYLBUR commands in the input are not executed.
NO PROFILE	Specifies that the profile is not to be executed, as it normally would be. This is useful when WYLBUR is being invoked as part of a TSO CLIST or batch job used by many different people.
UNNUMBERED	Specifies that line numbers are not being used and that all columns of each line are to be treated as input to WYLBUR. Normally, line numbers are stored in the WYLBUR command lines when the job is submitted. In that case, if the input is either submitted as part of the RUN command or is read from an F format data set, the contents of lines in the last eight columns are

ignored. If the input is a V or U format data set, the first eight columns are ignored.

OPTIONS are followed by an equal sign; the options are enclosed in single quotation marks and separated by commas.

Example

```
// EXEC WYLBUR,OPTIONS='NO ABORT,  
// UNNUMBERED,ARGUMENT="ABC" '
```

The SET TABS and SET TERMINAL commands are not used when running WYLBUR as a batch job. If either is used, an error message will be displayed.

10.3.3 WYLBURL

The WYLBURL procedure executes WYLBUR commands as a batch job and permits the use of up to 50 active files, with each active file using the maximum amount of space. Use WYLBURL when several large active files are needed.

There is one optional symbolic JCL parameter available for the WYLBURL procedures: OPTIONS, which indicates options to be used by the WYLBUR program. The OPTIONS parameter is described in the WYLBUR section.

In the following example, WYLBURL is used to obtain a formatted listing of a document. The document will be printed without line numbers.

Example

```
// EXEC WYLBURL  
LIST FROM MANUAL ON CAT MARKER # UNNUMBERED
```

10.4 Sending Email from Titan

10.4.1 SENDMAIL

SENDMAIL is a batch procedure for sending email from Titan. The following rules apply:

- The MAILIN DD statement can be any data set, with LRECL less than or equal to 240 that contains the email lines.
- The 'FROM:' record is required and should specify one valid email address. Any delivery error messages will be sent to this address. Since Titan currently has no facility for receiving email, the email address should be on some other email system where you already receive and read your email.

-
- The 'TO:' record is required and should contain one or more email addresses separated by blanks, commas, or semicolons. In addition, multiple 'TO:' records can be specified.
 - The 'SUBJECT:' record is not required, but should normally be specified.
 - Additional email header records can be added:

```
cc: recipient@mail.nih.gov
reply-to: reply@mail.nih.gov
```

The 'CC:' record can contain one or more email addresses and there can be more than one 'CC:' record.

The 'REPLY-TO:' record, if used, should contain one email address to receive replies to the email.

Example

In the following example the headers and body of the mail are included in the job control language:

```
//S1 EXEC SENDMAIL
//MAILIN DD *
from: senderid@mail.nih.gov
to: recipient@mail.nih.gov
subject: mail test of SENDMAIL

body of email
more body
end of email
```

Example

In this example, there are two concatenated data sets, one containing the header, and one containing the body of the email.

```
//stepname EXEC SENDMAIL
//MAILIN DD=userid.headers,DISP=SHR
//          DD=userid.report,DISP=SHR
```

10.5 VISION:Builder (MARK IV)

VISION:Builder (MARK IV) can do complex file handling, database management, and report functions. It is available as a legacy product.

10.6 MAX Utilities

MAX is a suite of three software products designed for data manipulation in the OS/390 environment. The individual products are named MAX DATA/UTIL, MAX/BATCH, and MAX/PDF.

For a user guide and detailed information on MAX, go to:

<http://silkad.nih.gov/public/pcc.ejs.@www.max.htm>

10.6.1 MAX DATA/UTIL

This product is a data utility that functions exclusively in the area of VSAM and SAM (Sequential Access Method) data sets. The SAM portion came into being to fill the void created by ISPF/PDF's inability to edit records whose length was over 256 bytes. Even though IBM rectified that situation in Release 4.0 of ISPF, MAX DATA/UTIL still contains many useful SAM functions that have no counterpart in ISPF. Since ISPF/PDF still has no capacity in the VSAM area, software such as DATA/UTIL is a definite requirement for working with VSAM data sets.

In addition to providing browsing and editing capability in the VSAM and SAM areas, MAX DATA/UTIL provides the ability to select a subset of records from a file for viewing and updating. It also provides the means to selectively copy records from an input file to an output file or to multiple output files based on the content of each record. DATA/UTIL can reformat records on output; changing the size of the record if required.

DATA/UTIL provides online access to the IBM utility IDCAMS and its DELETE, DEFINE, ALTER, REPRO, BLDINDEX and VERIFY functions. It allows an existing VSAM data set to be used as a model for creating new data sets; adopting the parameters of the model but allowing individual parameters to be changed. It comes with an extensive help system that is sensitive to the cursor position on the screen. MAX processes the following IDCAM Entry types:

ALIAS	An entry that relates an alias (alternate entry name) to the real entry name in a user catalog.
AIX	An alternate index, conceptually a key-sequenced cluster. An alternate-index entry points to data and index components as well as a base cluster entry.
KSDS	Key-sequenced VSAM data set, includes data and index components.
ESDS	Entry-sequenced VSAM data set, contains data component only.
RRDS	Relative record VSAM data set, contains data component only.

VRDSD	Variable length RRDS VSAM data set, includes index component as well.
LINEAR	Linear VSAM data set, contains data component only.
GDG	An entry that permits non-VSAM data sets to be associated (to maintain an historical collection) with other non-VSAM data sets with the same name.
NONVSAM	A non-VSAM data set which may reside on tape or DASD.
PATH	An alias name that identifies an alternate-index and its base cluster.
UCAT	Pointed to by the master and is used to lessen the contention to the master catalog and to facilitate volume portability.

10.6.2 MAX/BATCH

MAX/BATCH performs most of the features that the online MAX DATA/UTIL does but in the batch mode. In addition to VSAM and SAM, PDS capability is added in the batch mode. Simple control statements permit the selective copying, updating and reformatting of data sets. MAX/BATCH can be incorporated into job streams to select records for further processing in subsequent job steps.

10.6.3 MAX/PDF

MAX/PDF is designed to enhance the ISPF/PDF Browse and Edit functions and expands on the data set information available in a VTOC listing. MAX allows the building of data set name lists (DSNLs) that enable the user to group together all of the data sets associated with a given project or particular task. These DSNLs are similar to the lists formed by ISPF/PDF option 3.4. MAX differs in that the user can tailor each DSNL to contain any specified data set name. The user is not confined to a list matching just one partially qualified DSN as with ISPF option 3.4. The DSNs on a DSNL are numbered and can be selected from the command line without the need to move the cursor to the particular line. In addition to being able to select the DSN by number the user can also supply a letter to indicate what they want to do with the DSN; e for edit, b for browse, etc.

Each user can have as many DSNLs as they wish and working from these lists saves time and effort. The keystrokes required to do a quantity of work are greatly reduced.

One useful feature of MAX/PDF is the ability to reduce the Member Selection List (MSL) of a PDS to one that contains only those members containing a user provided search criteria. For example, while looking at a MSL of a PDS, the user can reduce the MSL to only those members that contain the string "FORTRAN" by issuing a simple Find command. After reducing the MSL to those members containing the

criteria the members can be browsed or edited or the whole reduced MSL can be bulk edited or copied to another data set via simple commands.

Another useful feature of MAX/PDF is called "Point n Shoot." By placing the cursor in a data set name occurring anywhere on the screen and pressing a PF key, the users session is automatically transferred to a screen that has that data set name inscribed. The user can then edit, browse or obtain information about the characteristics of that data set. When finished, the session is returned to where the "Point n Shoot" was initiated. This feature allows a programmer to answer questions about data sets without interrupting their session and chain of thought.

11 Tape Storage

Most Titan customer tape data resides in the Virtual Storage Manager (VSM)—a comprehensive tape storage system consisting of tape silos, disk buffers, new tape technology, and tape management software that improves performance and reduces human intervention in storing, retrieving, and mounting tapes. All VSM tapes are numbered above 500000.

11.1 Tape Standards

The standards for the Titan tape storage system are as follows:

- The NIH tape library contains only virtual 3490 cartridge tapes with standard labels.
- All 9-track and 3480 tapes will be treated as foreign tapes. Refer to section 11.7 for information on foreign tape processing.
- Tape data set names must begin with a valid userid or RACF group (account).
- Tape data sets must have an expiration date or retention period.
- Tape data set security on Titan is handled by RACF permissions on a data set rather than volume basis. For additional information, see the RACF section of the Titan User's Guide.
- All tape data set names must conform to standards for DSNAME described in section 2.10.3, "Creating Permanent or Temporary Data Sets on Tape." Tapes that are not under the control of TMS must be processed as foreign tapes.
- Each data set on a tape should have the same userid as the first data set. The reason for this standard is that the userid of the first data set cannot be deregistered. A tape data set with a deregistered userid is unusable. If all the data sets have the same userid, they will all be usable.

11.1.1 Recreating Tape Data Sets

When recreating a data set on a specific cartridge tape volume, observe the following rules:

- For a tape with multiple data sets, only the last data set can be recreated.
- Recreation is not allowed if the data set has an expiration date of 99365.

-
- Do not change the data set name. Only when the data set name on the DD statement matches the data set name on the cartridge tape label will the expiration date be ignored and TMS allow the file to be overwritten.
 - Specify DISP=OLD. (DISP=NEW is used exclusively for requesting a scratch volume to create a new data set or adding a new data set to an existing cartridge tape volume).

If either of the last two conditions is not met, a scratch volume will be used, and the only indication that this has occurred is in the JES2 Job Log at the start of the job output.

11.1.2 Bypass Label Processing (BLP)

Bypass label processing (BLP), a subparameter of the LABEL=, is not allowed on Titan. Users processing foreign tapes must ensure that the tape has been created as non-labeled or with IBM standard labels. Users must also ensure that the supplier provides the data set name, density, record format, record size, and block size.

If any difficulties are encountered, contact the NIH Help Desk for further assistance.

11.2 CA-1 Tape Management System Features

Titan uses the CA-1 Tape Management System (TMS) software to maintain accountability of the tapes in the tape library. The system records the tape serial number and the data set name and characteristics in a tape management catalog when the tape is created. The catalog does the following:

- verifies that the proper cartridge tape is mounted
- produces comprehensive inventories of magnetic cartridge tape allocations by registered account group (agency) and userid

Data Set Expiration Dates and TMS

Users should become familiar with the TMS procedure for establishing and maintaining expiration dates of tape data sets. These procedures prevent inadvertent or premature release of tape data sets. Important points to be considered are:

- There is no "default" expiration date for tapes. Instead, users must specify an expiration date or retention period on the DD statement when they create or recreate a tape data set. **JCL that creates a new tape data set and does NOT include expiration information (EXPDT= or RETPD=) will FAIL.**
- The expiration date or retention period are specified on the DD statement. See the EXPDT and RETPD parameters of the DD statement in section 2.6.4.

-
- When you create a tape with multiple data sets, if you specify a numerically higher expiration date for any data set after the first one, then TMS updates the expiration for the first data set by setting it to the higher expiration date. This then becomes the controlling expiration date for the tape.
 - The expiration date can be extended or altered by other methods. See section 11.5 for details.
 - Multi-data set tapes are retained until all expiration dates on the tape have expired.
 - You can scratch a tape by setting the expiration date for the first data set on the tape to today's date. The tape will become scratch and eligible for reuse when the tape management process runs at about 6:00 am the next morning. This will work even if the tape has multiple data sets. When the tape becomes scratch, all data sets cataloged on that tape will be uncataloged.
 - Processing a data set for input does not affect its expiration date. However, output processing affects expiration dates in the following manner:
 - The expiration date is reestablished whenever a tape is used for output. The retention period must be specified on the DD statement.
 - If a tape is "OPENED" for output and is not successfully "CLOSED" by the processing program (i.e., it is closed by abnormal termination processing), the retention period for the tape will be five calendar days regardless of what may have been specified on the DD statement. This is based on the assumption that any data set closed by abnormal termination processing is going to be recreated later in a successful run. However, the five-day period allows the user time to take action to keep the data if it has some value. This applies both to creation of new data sets and to recreation of existing data sets.
 - The DISP=MOD parameter of the DD statement, used to append data to an existing data set, does not change the expiration date.

Tape Unit Names

The following unit names refer to the standard NIH tapes:

- TAPE, CTAPE - virtual 3490 standard tapes in the NIH tape library

The following unit names refer to incoming foreign tapes or tapes to be written outside of the VSM for removal from NIH. These tapes are not under control of the tape management system:

- FRGN6250, 9TRACKHI - 9-track 6250/1600 bpi tape unit (the default output density is 6250 bpi)
- FRGN3480 - 3480 cartridge tape unit.
- FRGN3490 - 3490 cartridge tape unit.

The following examples illustrate data sets with expiration date specifications. The cartridge tape will be retained until the 115th day of 2003. The first qualifier for the data set name can be either the tape owner's userid or the account group.

```
//DD1 DD UNIT=TAPE,DSNAME=userid.MYTAPe1,
// DISP=(NEW,CATLG),EXPDT=2003/115

//DD1 DD UNIT=TAPE,DSNAME=account.MYTAPe1,
// DISP=(NEW,CATLG),EXPDT=2003/115
```

The following examples illustrate data sets with retention period specifications. The cartridge tape will be retained for 130 days from the date of creation.

```
//DD2 DD UNIT=TAPE,DSNAME=userid.MYTAPe2,
// DISP=(NEW,KEEP),RETPD=130

//DD2 DD UNIT=TAPE,DSNAME=account.MYTAPe2,
// DISP=(NEW,KEEP),RETPD=130
```

11.3 TMS Error Conditions

Various system completion codes and informative messages can result during TMS processing. The more serious abnormal situations result in the job step being abnormally terminated by TMS with an ABEND code of the form *x*EC where *x* can be a number or a letter. See sections 12.3.1 and 12.3.2 for a description of the ABEND codes.

Another abnormal situation occurs when a scratch tape is requested for output and TMS does not accept the mounted tape as a scratch tape. TMS issues an "IECTMS3... volser IS NOT SCRTCH (errorcode)" message with an accompanying error code and requests another tape. If the situation resulted from improperly coded DD statements, the job will be terminated. See section 12.2 for more information on common CA-1 messages.

11.4 Tape Data Set Security

Tape data set security on Titan is handled by RACF permissions on a data set rather than volume basis. Data set protection will depend on the RACF profiles in place and will apply to any data set, regardless of whether it is on tape or disk. Users may need

to modify their RACF profiles to provide the correct level of access to tape data. For RACF processing on the Web, go to:

<http://titan.nih.gov>

and select RACF.

If an agency wishes to continue the use of the old North system tape naming conventions of aaa.iii.dataname, the RACF coordinator must create RACF profiles of the form aaa.iii.** and should permit the \$iii userid ALTER access to that profile. Optionally, the RACF coordinator can set the owner for the aaa.iii.** RACF profile as \$iii. This will allow the \$iii user to use RACF commands to permit other users access to data sets on tape. Unless these RACF permissions are given, the generic profiles currently in use for DASD will be used on Titan and some tape jobs may fail.

11.5 Tape Inventory Management

Users can access the tape database for tape management (displaying information, extending expiration dates, or scratching tape data sets) through ISPF Option L.5, the TI command (see section 6.1.3), the TAPEINFO batch procedure (see section 6.1.1), or the VOLLIST batch procedure (see section 6.1.2).

11.6 Removal of Tapes

Tapes in the VSM may not be removed from the NIH Computer Center. If you need to check out a tape that has a serial number above 500000, you must first copy it to a foreign 3490 or 3480 tape cartridge for removal. Users may provide their own 3490 or 3480 tapes or purchase them from CIT.

To purchase a tape from CIT, go to <http://silk.nih.gov/purchase-tape>. Each tape will receive a standard internal label with the volume serial numbers you specify and be checked in as a foreign tape so that you may write on it. You will be notified via e-mail when your tape(s) are available. Please ensure that all contact information on the web page, including your box number, is correct. Tapes will be placed in the specified output box the next Monday.

11.7 Foreign Tape Processing

A foreign (or special) tape is defined as a tape not under control of the CA-1 Tape Management System. The NIH Computer Center processes 9-track foreign tapes recorded at densities of 6250 and 1600 bpi, 3490 cartridge tapes, and 3480 cartridge tapes. There is no 7-track capability at the NIH Computer Center.

Foreign Tape Check-in

All foreign tape processing is done at the NIH Computer Center's Bethesda campus. Parklawn users can use the NIH courier service to send tapes to Bethesda.

Foreign tapes can be checked in for up to a week, to be returned to output boxes the Monday following check-in. To make the most efficient use of tape-drive resources, all foreign tapes are mounted at the Bethesda campus. Users at the Parklawn site can check tapes in at the CIT Offsite Distribution Center located in the Parklawn Building (Room 2B-70). Tapes will be returned to output boxes the Monday following tape check-in.

The courier service operates twice daily each weekday between the NIH Computer Center (Building 12A Room 1000) and the CIT Offsite Distribution Center at the Parklawn Building (Room 2B70). There will be a delivery arriving at Parklawn at around 8:00 a.m. and again around 2:00 p.m.

If you have any questions concerning these procedures, please call the NIH Help Desk.

Processing

The following procedures must be followed to process foreign tapes:

- A foreign tape must be registered at the NIH Computer Center on each day that it is to be processed. Each reel tape must be secured with a tape band. The band must have the volume serial number written on it. An external adhesive label (supplied by the NIH Computer Center) must be attached to the reel or cartridge and provide the following information:
 - volume serial number (same as written on the tape band or cartridge edge)
 - programmer name
 - agency identifier code or account number
 - office mailing address
 - telephone number
 - tape density (6250 or 1600 bpi for 9-track, 3480 or 3490 for cartridge tape)
- When allocating a tape drive for mounting a foreign tape, the user must specify one of the generic names for foreign tapes that are provided in the UNIT parameter of the DD statement: The following unit names refer to incoming foreign tapes or tapes to be written outside of the VSM for removal from NIH. These are for tapes not under control of the tape management system:

-
- FRGN6250 -- 9-track 6250/1600 bpi tape unit (the default output density is 6250 bpi)
 - FRGN3480 -- 3480 cartridge tape unit.
 - FRGN3490 -- 3490 cartridge tape unit.
 - The DD statement allocating a tape drive for a foreign tape must also include the parameter EXPDT=98000.

12 Messages and Codes

The following section describes messages and codes for Titan.

12.1 Common ABENDs

When computing tasks end abnormally (called an ABnormal END, or ABEND), the system provides ABEND messages that describe the problem, with return codes that further identify the problem. Here is a list of some common ABEND codes along with brief descriptions of their meanings for typical return codes.

x13 ABEND Codes

ABEND Code	Description
x13	(where "x" is a number or a letter up to F) The batch job or interactive program could not open a data set needed by the job.
013	Conflicting or unsupported parameters in DCB: member name specified in DD not found; no directory allocation subparameter in DD. Refer to the explanation of message IEC141I for complete information. Possible Causes <ul style="list-style-type: none">• DCB parameters conflict, no BLKSIZE or not a valid multiple of LRECL• missing or misplaced SYSIN DD statement• partitioned data set member not found• JCL - tried to create a PDS without allocating directory blocks
213	Unable to locate a data set (and thus was also unable to open it). Perhaps the name was specified incorrectly or the wrong data storage volume was indicated in the JCL of the job. The solution is to find (or create) the data set needed by the job and to make sure the JCL points to the data set correctly. Refer to the explanation of message IEC143I for complete information.
613	Unable to open a tape data set due to an I/O (input/output) error on the tape. The data set could not be read from or written to the tape as requested by the job. The solution is to find out what is on the tape via TAPEINFO (or TAPEMAP for a foreign tape) and ensure that the job isn't trying to do something impossible (like write a 5th data set on a tape that doesn't contain a 4th data set). If a flaw in the magnetic recording media of the tape is causing this problem, assistance might be required. Contact the

NIH Help Desk if there is a problem. Refer to the explanation of message IEC147I for complete information.

813 Tried to open a tape data set, but the data set name indicated in the JCL doesn't match the data set name on the tape. Solution: find out the real name of the tape data set and correct the JCL. Refer to the explanation of message IEC149I for complete information.

x22 ABEND Codes

ABEND Code	Description
222	The operator or JES2 cancelled the job. Check the job log listing for more information. If there is no apparent explanation, contact the NIH Help Desk before resubmitting.

Possible Causes

- failed to free a dataset needed by your job
- multiple jobs in execution after the same tape
- JCL error caused mount request for invalid or non-existing volume
- requested devices not currently available (job may be re-queued)

322 Job or step time exceeded the specified limit.

Possible Causes

- program is in a loop
- insufficient time parameter on JOB statement

722 The batch job tried to produce more lines of output than requested in the JOB statement or, if no output lines request was specifically coded in the JOB statement, exceeded the default amount of output lines for the JOB.

The solution for this ABEND is to ensure that you want to print all the output in question and, if so, to increase the output lines request in the LINES parameter of the /*JOBPARM statement.

x37 ABEND Codes

ABEND Code**Description**

B37

The batch job or interactive program couldn't complete output to a data set because the amount of disk space allocated for the data set wasn't sufficient. The solution is to make sure that the output disk data set is given a large enough primary allocation. It is advisable to make appropriate allocations in the primary space allocation.

Possible Causes

- the data set on DASD output already had 16 extents, but required more space
- secondary space was too small
- no more space was available on the volume
- the volume table of contents was full

D37

Partitioned dataset is full.

Possible Causes

- 100% full (check: ISPF 3.2)
- directory blocks all used up (check: ISPF 3.2)

0Cx ABEND Codes**ABEND Code****Description**

0C1

The program received an invalid operation code, usually the result of a "wild branch" in the program or a missing DD statement in the JCL.

Possible Causes

- subscript error – overwrote existing code
- tried to read a file that was not opened
- misspelled DDNAME
- error in parameters passed to subroutines
- missing DD statement
- recording mode was wrong or density was incorrect
- bad load module, possible bad object deck

0C4

The program tried to write in an area of memory that doesn't belong to the job, also called a "protection exception."

Frequently this is the result of the job trying to write to an array location that is greater than the size defined for the array. The solution is to check the program to find out where the array location is going out of bounds.

Possible Causes

- tried to execute code outside of the program
- bad branch
- subscript error
- tried to move data beyond the end of a buffer, or getmain area; overlaid instruction(s)
- invalid address was referenced due to subscript error or bad parameters
- the key of an instruction or an operand in storage does not match the protection key in the PSW (interrupt code 4)
- the virtual segment or page was never allocated (interrupt code 10 or 11, respectively)
- the page was paged out and the routine requesting the code was disabled for I/O interrupts (interrupt code 11)
- COBOL errors
 - in group move, receiving wrongly-defined record variable length
 - tried moving variable length record larger than target field maximum
 - tried to read or write a file which was not open
 - misuse of DD dummy statement
 - tried to call within COBOL's internal F SORT input/output procedure
 - used with SORT – tried to go back in the output procedure

0C7

The program (usually in COBOL) attempted to put incorrect data into a packed decimal field, resulting in this data exception ABEND. The solution is to ensure that the data is in the format that the program was written to handle.

Possible Causes

- Data exception: decimal data is incorrect or improperly overlapped or not validly initialized.

- Subscript error: expecting to find packed data, but packed data is not present; could be binary or EBCDIC; or table was not initialized.
- Failure to initialize a variable or array.
- subscript error, referenced beyond table
- COBOL errors
 - working storage not initialized
 - bad data, check data for errors
 - period missing after imperative statements within at end clause
 - binary field in an arithmetic operation is not large enough to accept result

80A ABEND Code

ABEND Code	Description
80A	The batch job or interactive program ran out of memory (REGION). Usually this can be remedied by increasing the REGION request for the job or job step that ABENDED. Section 9.2.1 contains information on how to determine the approximate amount of region needed.

The meaning of ABEND messages and return codes can be found in the IBM manuals available in the System Error Messages section of the CIT Publications web page (see section 1.3).

If you have questions about an ABEND received by a batch job or need assistance correcting an ABENDING batch job or interactive program, contact the NIH Help Desk.

12.2 ThruPut Manager Messages

DTM7108I PLEASE FREE DATA SET dsname

Explanation: You are holding a data set required by a batch job.

User Response: Free the data set if possible.

DTM7109I THIS DATA SET IS REQUIRED BY jobnumber jobname

Explanation: This message accompanies message DTM7108I.

User Response: Free the data set if possible.

DTM7110I jobname IN DCS status (nnn)

REQ SERVICE DSNAME

req STANDBY dsname

Explanation: status: HOLD or WAIT
nnn: The number of unavailable data sets
req: EXC for exclusive or SHR for shared
dsname: The data set name

The job is waiting for exclusive or shared access to a data set.

User Response: None. If the problem persists, run the job using EXC DRAIN.

12.3 TMS ABEND Codes and Messages

The ABEND codes and messages generated by the Tape Management System are provided in BookManager (select CA-1) and MVS/QuickRef. To access these tools on MVS, select C for Products on the CIT/Titan Primary Option Menu. Some of the more common job log messages are shown in section 12.3.3.

12.3.1 TMS ABEND Codes for IEFTMS50 Messages

During normal processing, certain error conditions may be noted. When an error condition exists, a system generated ABEND is invoked. The following ABENDs generate IEFTMS50 messages:

1EC - Invalid Data Set Name

Explanation: The data set name in the JCL does not agree with the data set name in the Tape Management Catalog (TMC) for the volume involved.

User Response: Verify that the data set name in the JCL is correct or the correct volume was requested.

2EC - Unable to Process a Secondary Data Set

Explanation: The user is trying to open the second or higher data set on a tape

volume.

User Response: If the tape is being opened for input processing, verify that the JCL contains the correct file sequence number and the correct volume was requested. If the tape is being opened for output processing, the message IECTMS3 contains a reject code that describes the error. (See section 12.3.3.)

3EC - Invalid Multivolume or DSNB Pointers and Specification Errors

Explanation: An invalid volume serial number or DSNB pointer has been found.

User Response: Make sure the volume serial number is entered correctly. If it is correct, contact the NIH Help Desk.

4EC - EOF - Input Only

Explanation: The multi-volume file being opened for input is not the same volume that the data set was created on.

User Response: Verify that the correct volume serial number was specified in the JCL.

5EC - Mixed Expiration Date Disallowed

Explanation: Mixed expiration dates are not allowed.

User Response: Provide correct expiration dates in the JCL.

6EC - Invalid Password

Explanation: An attempt was made to open a password protected data set and the user did not supply the correct password.

User Response: Supply the correct password.

7EC - Dismount for which a Remount Will Not Satisfy the Request

Explanation: A MOD tape has been rejected.

User Response: See the IECTMS3 message in section 12.3.3 for the condition

code that describes the error.

12.3.2 TMS ABEND Codes for IEFTMS70 Messages

During normal processing, certain error conditions may be noted. When an error condition exists, a system generated ABEND is invoked. The following ABENDs generate IEFTMS70 messages:

1EC - IPL Required

Explanation The TMC has been extended or moved.

User Response: Contact the NIH Help Desk.

2EC - All Data Set Name Blocks Are Allocated

Explanation: There are no secondary data set name blocks (DSNBs) that can be allocated for this request.

User Response: Contact the NIH Help Desk.

3EC - Invalid DSNB Record

Explanation An invalid DSNB or control record has been found.

User Response: Contact the NIH Help Desk.

4EC - Invalid Parameter List

Explanation: The parameter list passed to SVC ED contains an invalid field.

User Response: Verify that no user modification is causing the ABEND.

7EC - Backup is needed immediately

Explanation: The audit data set is full.

User Response: Contact the NIH Help Desk.

DEC - I/O Error has Occurred

Explanation: An I/O error has been detected on the audit data set.

User Response: Contact the NIH Help Desk.

EEC - I/O Error Has Occurred

Explanation: An I/O error has been detected on the TMC data set.

User Response: Contact the NIH Help Desk.

12.3.3 TMS Messages in the JES2 Message Log

nn IECTMS1 ddd, ,ENTER VSN

Explanation: An NL or BLP tape is being opened for input or output.

User Response: Reply with the correct volume serial number. If a specific mount request was made, only one matching reply is required. If a non-specific request was made two successive, matching replies are required.

nn IECTMS2 ddd,vvvvvv,VERIFY TAPE FROM OUTSIDE LIBRARY

Explanation: A duplicate volume serial number in the TMC range is being opened for output and the JCL specifies LABEL=EXPDT=98000.

User Response: Reply 'M' to unload the tape so the correct tape can be mounted. Reply 'U' to verify that the tape is indeed from outside the user's library and is not under TMS control.

IECTMS3 ddd,vvvvvv IS NOT SCRATCH (xx)

Explanation: IECTMS3 A tape mounted as 'scratch' is unacceptable as 'scratch' to TMS.

User Response: Mount another tape as 'scratch'. The tape is unloaded. A tape may be determined not to be 'scratch' for a number of reasons. The condition code (xx) in the IECTMS3 message indicates the reason for rejecting the 'scratch' mount. The condition codes are defined below:

04 = Operator replied 'M' to a '98000' request (see IECTMS2)

08 = Volser not in the TMC and EXPDT = '98000'

12 = Operator replied 'M' to a '98000' EXPDT check (see IECTMS4)

16 = Operator did not reply a specific NL volser (see IECTMS1)

- 20 = TMC EXPDT=99365 (permanent hold) cannot be recreated
- 24 = Volume marked OUT-OF-AREA
- 28 = TMS data set name does not match tape label
- 32 = Volume has not expired (not processed by TMSCLEAN)
- 36 = Same volume remounted for EOV output request
- 40 = Volume part of valid multi-volume chain
- 44 = Specific volser required (see IECTMS7)
- 48 = Volume part of valid multi-volume chain (trying to create 1st file)
- 52 = Trying to create file 'N' and 'N-1' does not exist
- 56 = Trying to create file 'N' and 'N+1' already exists
- 60 = Trying to create file 'N' but 'N' already exists
- 64 = Trying to recreate a data set with DISP=NEW
- 68 = Trying to change label type or density on file sequence greater than 1
- 76 = Trying to recreate data set with RETPD=0

nn IECTMS4 ddd,vvvvvv IS UNEXPIRED

Explanation: The expiration date in the HDR1 record of an output tape not controlled by TMS (i.e., XPDT=98000) is greater than the current date.

User Response: Reply 'M' to dismount the tape. Reply 'U' to override and write on the tape.

IECTMS6 ddd,vvvvvv IS APPROVED FOR LABEL CHANGE

Explanation: A TMS controlled tape has been approved for label change. The console operator can re-label the volume in response to OS messages.

User Response: None.

IECTMS7 ddd,vvvvvv, SPECIFIC REQUEST

Explanation: A specific request for volume VVVVVV is being made at EOV.

User Response: Mount the requested tape on drive DDD.0

12.4 ABR Online Messages for Data Set Retrieval

FDRW70 ELIGIBLE FOR RECALL --DSNAME.userid.XXXXXX.DATA

FDRW71 TYPE 'UC' TO UNCATALOG THE DATA SET

FDRW71 TYPE 'YES' TO PERMIT THE RESTORE OR PRESS 'ENTER' TO BYPASS

Explanation: The automatic recall function is in effect for online processing.

User Response: Press the 'ENTER' key to cancel the recall or 'YES' to restore.
Do not enter 'UC' to uncatalog the data set.

FDRW76 DATA SET IS SCHEDULED TO BE RESTORED TO VOLUME SERIAL NUMBER
PUBnnn

FDRW77 TYPE NEW VOLUME SERIAL NUMBER AS XXXXXX OR PRESS 'ENTER' TO
CONTINUE

Explanation: The recall function permits the data set to be restored to the
original storage volume or to another volume.

User Response: Enter a new storage volume number or press the 'ENTER' key
to restore the data set to its original volume.

FDRW78 RESTORE START COMMAND ISSUED -- YOU WILL BE NOTIFIED UPON
COMPLETION

Explanation: While the data set is being restored, the terminal is available to
perform other tasks.

User Response: Proceed to perform other tasks while waiting for the restore to
complete.

FDRW311 FOR RESTORED DSNAME=userid.XXXXXX.DATA ALLOCATED CATALOGED

Explanation: The data set is ready for immediate use.

User Response: None Note: In batch mode, jobs simply pause and wait for the
data set to be restored before continuing to execute.

Note: In batch mode, jobs simply pause and wait for the data set to be restored before
continuing to execute.

INDEX

- () parameter of JOB, 7
- * parameter of DD, 34
- /*AFTER, 14
- /*BEFORE, 15
- /*BOX, 16
- /*CNTL, 16
- /*DISCOUNT, 17
- /*JCS BATCH, 17
- /*JOBPARM, 18
- /*NETACCT, 19, 94
- /*OUTPUT, 20
- /*PASS, 20
- /*PRIVATE, 20
- /*ROUTE, 20
- /*ROUTE XEQ translated, 21
- /*SIGNOFF, 95
- /*SIGNON, 95
- /*WITH, 21
- /*XMIT, 22, 94, 212
- //*+JBS BIND, 22
- 9TRACKHI unit name, 246
- ABENDs, 251
 - dedicated disk access, 188
 - manuals, 255
 - S9EC, 204
 - TMS, 256, 258
- ABR messages, 261
- access method services, 181
- accounting information, 7
- ACCT parameter of EXEC not supported, 26
- address space, 61
- ADSERASE procedure, 129
- ADSMAP procedure, 185
- ADSRECOV procedure, 193
- agent parameter of JBS BIND, 23
- allocating space, 52
- AMASPZAP restricted, 108
- AMP parameter of DD, 35
- ASCII data, 45, 125
- assistance, 2
- AVGREC parameter of DD, 35
- backing up data sets
 - DDSSAVE, 142
 - deleting backups, 183
- DISKSAVE, 191
- DSSAVE, 116
- EDSSAVE, 157
- listing backups, 187
- PDSSAVE, 174
 - policy for dedicated disks, 188
- BACKLIST procedure, 187
- batch name conventions, 18
- BATCHTSO, 234
- BDAM data sets, 43, 54
- Binder, 226
- BLKSIZE subparameter of DCB, 42
- block size, recommended, 42
- BLP, 244
- BookManager, 3, 256
- BUFNO subparameter of DCB, 44
- Bypass Label Processing, 244
- canceling a TSO session, 234
- canceling jobs, 92
- carriage control, 80, 85
- cataloged procedures, 68. *See also* utilities
 - adding DD statements, 71
 - assigning values, 70
 - modifying EXEC parameters, 70
 - nullifying symbolic parameters, 70
 - overriding DD statements, 71
- cataloging data sets, 120
- CATDS procedure, 123
- CC option of LIST command, 114
- Center for Information Technology, 1
- character sets for printing, 82
- charges, reducing, 42
- CHARS parameter of DD, 35
- CHARS parameter of OUTPUT JCL, 59
- CHKPT parameter of DD not supported, 33
- CKPTLINE parameter of OUTPUT JCL not supported, 58
- CLASS parameter of JOB, 7
- CLASS parameter of OUTPUT JCL, 59
- COBOL, 225
- codes, completion, 120
- combining data sets, 139, 141

comments in JCL, 60
 COMPARE procedure, 124
 comparing data sets, 124
 compilers
 COBOL, 225
 FORTRAN, 225
 PL/I, 226
 completion codes, 120
 compressing dedicated disks, 189
 concatenating data sets, 139, 141
 COND parameter of EXEC, 28
 COND parameter of JOB, 8
 condensing a PDS, 170
 condition codes, 120
 CONNECT:Direct, 216
 contention, data set, 102
 continuous form paper, 76
 conventions, 2
 conventions, batch name, 18
 COPIES parameter of DD, 35
 COPIES parameter of OUTPUT JCL, 59
 COPY command of EDSUTIL, 112
 LIST option, 113
 RECFM U option, 113
 SKIP option, 113
 TAKE option, 113
 COPY procedure, 125
 courier service, NIH/Parklawn, 248
 creating data sets, 64
 CTAPE unit name, 245
 Custom ISPF panels, 232
 customer support, 2
 cut-sheet printing, 78
 Data Definition statement, 32
 DATA parameter of DD, 34
 data sets
 allocating space on disk, 52
 cataloging, 120
 comparing, 124
 concatenating, 139
 contention, 102
 copying, 125
 creating, 63, 64
 creating permanent on tape, 65
 creating temporary on disk, 63
 creating temporary on tape, 65
 deleting, 64
 deleting backups, 183
 generation, 161
 JCL for using, 66
 limits on tape, 65, 120
 list backups for data sets, 187
 list names of data sets, 182, 185, 199
 management class, changing, 183
 managing with ISMF, 181
 merging, 141
 passed, 67
 printing, 131
 renaming on disk, 137
 restoring lost, ADSRECOV, 193
 restoring lost, ISMF, 184
 retrieving, 120
 saving, 120
 scratch space limit on disk, 61
 scratching on disk, 137, 138
 sorting, 139
 space used, totaling, 183
 uncataloging, 120
 using, 63
 utilities for tapes, 197
 DATACLAS parameter of DD, 35
 DCB parameter of DD, 41
 DCB subparameters, 42
 DD statement, 32
 creating data sets, 64
 creating temporary data sets, 63
 overriding or adding, 71
 passed data sets, 67
 STEPLIB, 68
 DDNAME parameter of DD, 46
 DDSCOPY procedure, 146
 DDSGET procedure, 144
 DDSLIST procedure, 146
 DDSSAVE procedure, 142
 deallocating data sets, 49
 dedicated disks. *See* disks, dedicated
 DEFAULT parameter of OUTPUT JCL, 59
 delays, avoiding, 102
 deleting data sets, 64
 DEN subparameter of DCB, 46
 density of tapes, 46

DEST parameter of DD, 47
 DEST parameter of OUTPUT JCL, 59
 device names for tapes, 123
 DFSORT program, 139
 direct access data sets, 142
 discount processing, 17, 23
 DISKEXPT not supported, 108
 DISKGET procedure, 189, 192
 DISKMAP procedure, 189, 190
 disks, dedicated

- backing up, 191
- compressing, 189
- DISKGET procedure, 192
- DISKMAP procedure, 190
- DISKSAVE procedure, 191
- errors, 188
- maintenance and policies, 188
- restoring from tape, 192
- VTOC map, 190

 DISKSAVE procedure, 191
 DISP parameter of DD, 47
 DLM parameter of DD, 47
 DMBATCH utility, 217
 documentation, 1, 2
 DOUBLE option of LIST command, 115
 DPRTY parameter of EXEC not supported, 26
 DSCOPY procedure, 127
 DSF not allowed, 108
 DSGET procedure, 118
 DSID parameter of DD not supported, 33
 DSLIST procedure, 131
 DSNAME parameter of DD, 47
 DSORG subparameter of DCB, 44
 DSRENAME procedure, 137
 DSSAVE procedure, 116
 DSSCM not supported, 108
 DSSCMI not supported, 108
 DSSCR procedure, 137
 DSSCRV procedure, 138
 DSSUBMIT procedure, 220
 DSSUBMIT program, 221
 DSUNSCM not supported, 108
 DSUNSCMI not supported, 108
 DTM7108I, 103
 DTM7109I, 103
 DTM7110I, 102
 DUMMY parameter of DD, 48
 duplex printing, 78
 EDIT format data sets

- defined, 148
- EDSGET, 158
- EDSIN, 148
- EDSLIST, 153
- EDSOUT, 150
- EDSSORT, 154
- EDSUTIL program, 148
- reading, 148
- writing, 150

 EDSGET procedure, 158
 EDSIN procedure, 148
 EDSLIST procedure, 153
 EDSOUT procedure, 150
 EDSSAVE procedure, 120, 157
 EDSSORT procedure, 154
 EDSUTIL procedure, 115
 EDSUTIL program, 111, 148

- COPY command, 112
- EDSUTIL procedure, 115
- LIST command, 114

 efficiency, improving, 42
 email, 238
 error messages, 251

- ABENDs, 251
- ABR, 261
- completion codes, 120
- FDRWxxx, 261
- IECTMSx, 259
- IEF645I from JCLSCAN, 220
- JES2 message log, 259
- TMS, 256, 258

 EXEC statement, 26

- modifying parameters, 70

 EXPDT parameter of DD, 48
 expiration dates for tapes, 48, 50, 244
 FCB. *See* forms control buffers
 FCB parameter of DD, 49
 FCB parameter of OUTPUT JCL, 59
 FETCH procedure, 221
 file transfer

- CONNECT:Direct, 216
- host-to-host, 211

- to and from remotes, 207
- first-in, first-run, 18
- fonts, 82
- foreign tapes, 247
- FORMDEF parameter of OUTPUT JCL, 59
- forms
 - cut-sheet laser printing, 78
 - impact printing, 80
 - standard printing, 76
 - user-supplied, 81
- forms control buffers
 - Computer Center FCBs, 86
 - label FCBs, 81
 - user designed FCBs, 87
- FORMS parameter of OUTPUT JCL, 59
- FORTRAN, 225
- FREE parameter of DD, 49
- FRGN3480 unit name, 246
- FRGN3490 unit name, 246
- FRGN6250 unit name, 246
- GDG, 161
- generation data sets, 161
 - creating new GDG, 163
 - deleting, 163
 - model data set label, 162
 - VDSUTIL, 163
- GROUP parameter of JOB, 8
- hardcopy utility, 89
- help, 2
- hexadecimal tables for character sets, 83
- HOLD parameter of DD, 49
- host-to-host file transfer, 211
- HRECOVER ISMF command, 184
- IBM Session Manager, 235
- IBM utilities, 107
 - DFSORT, 139
 - IEBGENER, 125
 - IEHPROGM, 162
- ICAPRTBL not allowed, 108
- IDCAMS, 107, 181
- IEBCOMPR, 108
- IEBCOPY, 107, 108
- IEBDG, 107, 108
- IEBEDIT, 108
- IEBGENER, 107, 108, 125
- IEBIMAGE not allowed, 108
- IEBISAM, 108
- IEBTPCH, 107, 108
- IEBUPDTE, 107, 108
- IEFBR14, 136
- IEFTMS50 9XX- 12, 204
- IEHATLAS not allowed, 108
- IEHINITT not allowed, 108
- IEHLIST restricted, 108
- IEHMOVE restricted, 108
- IEHPROGM, 107, 162
- IEHPROGM restricted, 108
- IFHSTATR not allowed, 108
- impact printer, 80, 81
- INCLUDE groups, 74
- INDENT option of LIST command, 114
- input stream data sets, 34
- instream procedures, 73
- Interactive Output Facility, 95
- interactive processing
 - canceling jobs, 92
 - submitting jobs, 90
- Interface, 1
- internal reader, 220
- IOF, 95, 96, 101
- ISAM data sets not supported, 45
- ISMF, 181
- ISPF
 - Custom panels, 232
- JBS BIND statement, 17, 18, 22
- JCL
 - comments, 60
 - JCLSCAN procedure, 219
 - libraries, 68
 - order of statements, 6
 - overview, 5
 - restrictions, 62
 - scanning for errors, 219
 - syntax checking, 93
 - temporary data sets, 63
- JCLLIB statement, 24, 68, 74
- JCLSCAN procedure, 219
- JCS BATCH statement, 17
- JES2
 - control statements, 13
 - internal reader, 220

- job log, 103
- message log, 259
- RJE control statements, 95
- JESDS parameter of OUTPUT JCL, 59
- job
 - cancelling, 92
 - delays, 102
 - execution order, 16, 18
 - output explained, 103
 - running at another node, 93
 - saving output, 99
 - standards, 60
 - submitting, 90, 220, 221
 - to recall data sets, 136
 - turnaround, 63, 76
- job control language. *See* JCL
- Job Entry Subsystem. *See* JES2
- JOB statement, 6
- LABEL parameter of DD, 49
- labels, printing, 80
- language compilers, 225
- laser printers, 76
- limits for batch jobs, 61
- LINES parameter of JOB, 8
- lines per job, maximum, 61
- LIST command of EDSUTIL, 114
 - CC option, 114
 - DOUBLE option, 115
 - INDENT option, 114
 - MARKER option, 115
 - SPACING option, 115
 - TRIPLE option, 115
- LIST option of COPY command, 113
- Loader, 228
- LOGON panel, 231
- LRECL subparameter of DCB, 42
- magnetic tapes. *See* tapes
- management class, 51
 - changing, 122, 183
 - default, 57
 - setting, 56
- manuals, 1, 2
- MARK IV software, 239
- MARKER option of LIST command, 115
- MAX DATA/UTIL utility, 240
- MAX utility, 240
- MAX/BATCH utility, 241
- MAX/PDF utility, 241
- maximum limits for batch jobs, 61
- MEMADD procedure, 167
- MEMALTER not supported, 108
- MEMRENAM not supported, 108
- MERGE procedure, 141
- merging data sets, 141
- messages, 251
 - ABENDs, 251
 - ABR, 261
 - DTM7108I, 103
 - DTM7109I, 103
 - DTM7110I, 102
 - IEFTMS50 9XX- 12, 204
 - JES2 message log, 259
 - TMS, 256
- MGMTCLAS parameter of DD, 51
- MSGCLASS parameter of JOB, 8
- MSGLEVEL parameter of JOB, 9
- MVS/QuickRef, 256
- naming conventions for data sets, 121
- Network Job Entry, 211
- network printing services, 89
- NIH Help Desk, 2
- NIH/Parklawn Courier Service, 248
- NIHnet printing services, 89
- NJE, 211
- Node, 93
- Node names, 211
- NOTIFY parameter of JOB, 9, 93
- operating system, 5
- OPTCD subparameter of DCB, 44, 87
- order of execution, 16, 18
- ORDER parameter of JCLLIB, 25
- OUTLIM parameter of DD, 52
- output
 - boxes, 75
 - delivery, 75
 - printing, 75
 - saving, 99
 - turnaround, 63, 76
 - understanding, 103
- OUTPUT JCL statement, 57, 58
- OUTPUT parameter of DD, 52

PAGEDEF parameter of OUTPUT JCL, 60

PAGES parameter of JOB, 9

PALTAPE not supported, 108

Parklawn site courier to NIH, 248

PARM parameter of EXEC, 30

partitioned data set. *See* PDS

PASBILL not supported, 108

passed data sets, 67

PASSWORD parameter of JOB, 9

PDS

- add members, 167
- allocate space, 165
- condense, 170
- copy entire disk data sets, 171
- copy members, 173
- delete members, 166
- list contents, 169
- unload entire, 165, 174
- using as a procedure library, 25
- utilities to manage, 165

PDSALLOC procedure, 165

PDSCOPY procedure, 171

PDSDEL procedure, 166

PDSGET procedure, 175

PDSLIST procedure, 177

PDSMAP not supported, 108

PDSPACK procedure, 170

PDSSAVE procedure, 165, 174

PERFORM parameter of EXEC not supported, 26

PERFORM parameter of JOB not supported, 7

performance, improving, 42

PGM parameter of EXEC, 27

PL/I, 226

POSTSCRIP not supported, 108

PRINT procedure, 132

print trains, 82

printing

- 3160 printers, 78
- 3900 printers, 76
- character sets, 82
- continuous form paper, 76
- data sets, 131
- distribution, 75
- fonts, 87
- hardcopy utility, 89
- hexadecimal, 134
- ISO/ANSI carriage control, 85
- job turnaround, 76
- jobs from another node, 94
- labels, 80
- laser printing, 87
- machine carriage control, 85
- multiple fonts, 87
- network, 89
- output delivery, 75
- registering user forms, 81
- standard service, 76
- table reference characters, 87
- user-supplied forms, 81
- VTAM printers, 90

PrintShare, 89

private disks. *See* disks, dedicated

PROC parameter of EXEC, 28

procedure libraries, 24, 68

procedure-name parameter of EXEC, 28

procedures not supported, 108

procedures, cataloged, 68

Program Manager Binder, 226

programmer-name parameter of JOB, 10

PRTY parameter of JOB not supported, 7

PTFORM not supported, 108

publications, 1, 2

purchase tapes, 247

purging jobs, 61

QNAME parameter of DD not supported, 33

QuickRef, 256

RACF for tapes, 246

RCVFILE, 211

RD parameter of EXEC not supported, 26

RD parameter of JOB not supported, 7

recalling data sets via batch, 136

RECFM subparameter of DCB, 42

RECFM U option of COPY command, 113

reconnecting a TSO session, 233

record format, 80, 84

recovering data sets. *See* restoring data sets

reel drives for foreign tape, 248

region

- estimating size, 227
- limits, 61

REGION parameter of EXEC, 30

REGION parameter of JOB, 10

registering user forms, 81

Remote Job Entry workstations. *See* RJE workstations

renaming disk data sets, 121, 137

resources, 23

RESPOUT not supported, 108

RESTART parameter of JOB, 10

restoring data sets

- ADSRECOV, 193
- DDSGET, 144
- DISKGET, 192
- DSGET, 118
- EDSGET, 158
- PDSGET, 175
- using ISMF, 184

restrictions

- on JCL, 62
- on utilities, 108

retention period for tapes, 51, 52

RETPD parameter of DD, 52

return codes, 120

RJE control statements, 95

RJE workstations, 90

- canceling jobs, 92
- submitting jobs, 95
- transmitting data to/from tapes, 207

RMTDSIN procedure, 207

RMTDSOUT procedure, 207, 209

ROOM parameter of OUTPUT JCL, 60

S9EC ABEND, 204

scanning JCL, 219

scratching

- disk data sets, 137, 138
- multiple disk data sets, 121
- tape data sets, 203

screen, scrolling, 235

Secure Internet LinKed technologies, 90

security for tapes, 246

SENDFILE, 211

SENDMAIL, 238

Session Manager, 235

SILK batch job submission, 90

SKIP option of COPY command, 113

SNA network printing, 90

SORT procedure, 139

SPACE parameter of DD, 52

space, allocating, 52

SPACING option of LIST command, 115

standard printing service, 76

standards

- batch jobs, 60
- restricted utilities, 108
- tapes, 243

STEPLIB DD statement, 68

submitting batch jobs, 90, 93, 220

SUBSYS parameter of DD not supported, 33

symbolic parameters, 69

- assigning values, 70
- nullifying, 70

SYSCHK DD name not supported, 33

SYSCHKEOV DD name not supported, 33

SYSOUT parameter of DD, 55

table reference characters, 87

TAKE option of COPY command, 113

Tape Management System. *See* TMS

TAPE unit name, 245

TAPECOPY not supported, 108

TAPEINFO procedure, 198

TAPEMAP procedure, 203

tapes

- 9-track 6250 BPI, 191
- CA-1 Primary Option Menu, 202
- changing expiration date, 203
- creating data sets, 65
- DCB, 46
- density, 46
- device names, 123
- expiration dates, 48, 50, 203, 244
- foreign, 247
- information, 198, 202
- inventory management, 247
- ISPF panel L.5, 202
- multi-data set, 120
- purchase, 247

recreating data sets, 243
 reel drives, 248
 removal, 247
 reports, 247
 retention period, 51, 52, 203, 244
 scratching, 203
 security, 246
 silos, 243
 standards, 243
 storage, 243
 TAPEINFO, 198
 TAPEMAP, 203
 TI utility, 202
 TMS, 244
 TMS error conditions, 246
 transmitting data via RJE workstations,
 207
 unit names, 245
 utilities, 197
 VOLLIST, 199
 TAPESCAN not supported, 108
 temporary data sets, 63
 ThruPut Manager, 13, 22, 101, 102
 TI utility, 202
 TIME parameter of EXEC, 31
 TIME parameter of JOB, 10
 Titan transition, 1, 107, 235
 TITLE parameter of OUTPUT JCL, 60
 TMS, 244
 completion codes, 256
 error conditions, 246
 JES2 message log, 259
 messages, 256
 TMUSER command, 101
 TRCs, 87
 TRIPLE option of LIST command, 115
 TSLIST profile member, 232
 TSLOGON profile member, 231
 TSO
 canceling a session, 234
 Custom ISPF panels, 232
 LOGON panel, 231
 moving a session, 233
 reconnecting a session, 233
 TSLIST profile member, 232
 TSLOGON profile member, 231
 TYPRUN parameter of JOB, 12
 UDF, 24, 101
 uncataloging data sets, 120
 UNCATDS procedure, 124
 unit names for tapes, 245
 UNIT parameter of DD, 56
 UNITNAME command, 57
 User Display Facility, 24, 101
 USER parameter of JOB, 13
 user-supplied forms, 81
 USRPDSC procedure, 173
 USRPDSL procedure, 169
 utilities
 ADSERASE, 129
 ADSMAP, 185
 ADSRECOV, 193
 BACKLIST, 187
 BATCHTSO, 234
 CATDS, 123
 COMPARE, 124
 CONNECT:Direct, 216
 COPY, 125
 DDSCOPY, 146
 DDSGET, 144
 DDSLIST, 146
 DDSSAVE, 142
 DISKGET, 189, 192
 DISKMAP, 189, 190
 DISKSAVE, 191
 DMBATCH, 217
 DSCOPY, 127
 DSGET, 118
 DSLIST, 131
 DSRENAME, 137
 DSSAVE, 116
 DSSCR, 137
 DSSCRV, 138
 DSSUBMIT, 220, 221
 EDIT format data sets, 148
 EDSGET, 158
 EDSIN, 148
 EDSLIST, 153
 EDSOUT, 150
 EDSSAVE, 157
 EDSSORT, 154
 EDSUTIL, 111, 148

FETCH, 221
generation data sets, 161
IBM, 107
IDCAMS, 107
IEBCOPY, 107
IEBDG, 107
IEBGENER, 107
IEBPTPCH, 107
IEBUPDTE, 107
IEFBR14, 136
IEHPRGM, 107
JCLSCAN, 219
MAX, 240
MAX BATCH, 241
MAX DATA/UTIL, 240
MAX/PDF, 241
MEMADD, 167
MERGE, 141
not supported, 108
PDSALLOC, 165
PDSCOPY, 171
PDSDEL, 166
PDSGET, 175
PDSLST, 177
PDSPACK, 170
PDSSAVE, 165, 174
PRINT, 132
RCVFILE, 211
RMTDSIN, 207
RMTDSOUT, 207, 209
SENDFILE, 211
SENDMAIL, 238
SORT, 139
TAPEINFO, 198
TAPEMAP, 203
TI, 202
UNCATDS, 124
USRPDSC, 173
USRPDSL, 169
VDSUTIL, 121, 162, 163
VOLLIST, 199
WYLABEL, 81, 135
WYLBUR, 235, 236
WYLBURL, 238
VDSUTIL procedure, 121, 162, 163
Virtual Storage Manager, 243
VISION:Builder software, 239
VOLLIST procedure, 199
VOLSTAT not supported, 108
VOLUME parameter of DD, 57
volume table of contents, 190
VPS, 89
VS FORTRAN, 225
VSAM data sets, 45, 181
 DSRENAME restriction, 137
 VDSUTIL procedure, 121
VSM, 243
VTAM printers, 90
VTOC map, 190
World Wide Web
 canceling jobs, 92
 ordering publications, 2
 submitting jobs, 90
WYLABEL procedure, 81, 135
WYLBUR
 commands executed in batch, 236, 238
 Edit format used, 148
 transition, 235
WYLBUR procedure, 236
WYLBURL procedure, 238



Titan Batch Processing

Document Evaluation

Is the Manual:

	YES	NO
Clear?	<input type="checkbox"/>	<input type="checkbox"/>
Well organized?	<input type="checkbox"/>	<input type="checkbox"/>
Complete?	<input type="checkbox"/>	<input type="checkbox"/>
Accurate?	<input type="checkbox"/>	<input type="checkbox"/>
Suitable for the beginner?	<input type="checkbox"/>	<input type="checkbox"/>
Suitable for the advanced user?	<input type="checkbox"/>	<input type="checkbox"/>

Comments:

Please give page references where appropriate. If you wish a reply, include your name and mailing address.

Send to: National Institutes of Health
Center for Information Technology
NIH Computer Center
Building 12A, Room 4011
12 South Drive MSC 5607
Bethesda, MD 20892-5607

FAX to: (301) 496-6905

IC or Agency:
Date Submitted:
Name (Optional):
E-mail Address:



DEPARTMENT OF
HEALTH AND HUMAN SERVICES
PUBLIC HEALTH SERVICE

National Institutes of Health
Building 12A, Room 1011
12 South Drive MSC 5605
Bethesda, MD 20892-5605

FIRST-CLASS MAIL/Priority
POSTAGE & FEES PAID
DHHS/NIH
Permit No. G-821

OFFICIAL BUSINESS
PENALTY for PRIVATE USE \$300

