



What's New?



What's New?

Note

Before using this information and the product it supports, be sure to read the general information under “Notices” on page 43.

Second Edition, Softcopy Only (February 2007)

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

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

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

© Copyright International Business Machines Corporation 2004, 2007. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this book.	vii
Terminology and citations	vii
How to send your comments	vii
Chapter 1. The future is now.	1
Chapter 2. What's new in Version 8 of DB2 UDB for z/OS?	5
More flexibility with SQL	5
SELECT FROM INSERT statement	5
Generation of unique sequential numbers for applications	6
Ability to alter identity column attributes.	6
Dynamic scrollable cursors	6
Scalar fullselects in SQL statements	6
Integrated XML publishing functions in DB2	6
Common table expressions in SQL statements	7
Recursive SQL.	7
CURRENT PACKAGE PATH special register	7
GET DIAGNOSTICS statement	7
Compare null values with the DISTINCT predicate	8
Improved security	8
New built-in functions for data encryption and decryption	8
New data encryption tool	9
Multilevel security with row-level granularity	9
Easier identification of system users	9
CLIENT_ACCTNG special register	9
CLIENT_APPLNAME special register	9
CLIENT_USERID special register	10
CLIENT_WRKSTNNAME special register	10
Session variables	10
Improved encrypted security in distributed computing environments	10
Enhanced compatibility with the DB2 family	10
Extended limits for names and SQL statements	11
Longer column names	11
Longer and more complex SQL statements	11
Extensions to SQL procedure statements.	11
Longer index keys and predicates	11
Greater number of tables joined in a single FROM clause	11
Fewer restrictions for column functions	11
Qualified columns in the INSERT statement	12
ORDER BY clause for the SELECT INTO statement	12
Expressions in the GROUP BY clause.	12
More than one DISTINCT keyword allowed in a single query	12
Additional input format for timestamp strings.	12
Explicitly defined ROWID columns are not required for large objects	12
Descriptions for plans and packages in the DB2 catalog	12
Implicit drop of declared global temporary tables at commit	12
Significant support for Unicode and long names in the DB2 catalog	12
Unicode	13
Long names	13
Unicode, EBCDIC, and ASCII columns in the same SQL statement	13
Network computing enhancements	13
Increased portability of applications through consistent access to DB2 family servers	13
Improved performance for remote queries	14
Improvements in connectivity	15
IBM z/OS Application Connectivity to DB2 for z/OS feature.	15

ODBC expands support for encoding schemes	16
Scalability and performance	16
64-bit virtual storage	17
Materialized query tables	18
Ability to use an index in more situations	18
Capability to index predicates that have mismatched data types	18
Stored variable-length index keys	18
Backward index scans for avoiding sort operations	19
Additional distribution statistics for improved optimization	19
Improved optimization for dynamic SQL	19
Improved trigger performance	19
More parallelism for sort operations	20
Performance enhancements for star join qualified queries	20
Multiple fetches and inserts allowed within a single SQL statement	20
4096 partitions in a partitioned table space	20
Greater resource control with stored procedures and user-defined functions	22
Reduced overhead costs for data sharing workloads	22
Reduced lock propagation in the coupling facility	22
Improved control for accounting aggregation	22
Improved package-level accounting	23
Reliability, availability, and serviceability	23
Greater availability and flexibility with online schema evolution	23
Change column types and lengths	24
Add columns to an index	24
Add, rotate, or rebalance partitions dynamically	24
Change the partitioning and clustering of the data in your tables	24
Greater data availability with data-partitioned secondary indexes	25
More flexibility and faster recovery with system-level point-in-time recovery	25
Improved utility functions	26
Online REORG utility	26
Online CHECK INDEX utility	27
LOAD and UNLOAD utilities	27
RUNSTATS distribution statistics	27
Autonomic restart	27
Enhanced LPL recovery processing	27
Change more parameters online without recycling DB2	28
Increased maximum number of active and archive log data sets	28
Autonomic space allocation	28
Chapter 3. Leveraging your enterprise for e-business	29
Accessing your enterprise data	29
DB2 QMF family of products	29
Highlights of Version 8 of DB2 QMF	29
DB2 QMF for WebSphere	30
DB2 QMF for Windows	31
DB2 QMF for TSO/CICS	31
DB2 QMF High Performance Option	32
New packaging options for DB2 QMF	32
Integrating your zSeries and iSeries data with DB2 Connect	32
Boosting application development productivity with DB2 Development Center	33
Customizing and extending your enterprise data with XML Extender	34
Managing your enterprise	35
Managing DB2 UDB for z/OS data sets from the DB2 Control Center	35
DB2 Administration Server	36
Using msys for Setup DB2 Customization Center to customize DB2	36
DB2 Visual Explain	37
DB2 Estimator	38
DB2 and IMS Tools for your database server	38
Glossary	41

Notices	43
Trademarks	44
Index	47

About this book

This book provides an executive overview of Version 8 of IBM® DB2 Universal Database™ for z/OS®. This book is written for people who are evaluating DB2.

Visit the following Web site for information about ordering DB2 books and
obtaining other valuable information about DB2 UDB for z/OS:
<http://www.ibm.com/software/data/db2/zos/library.html>

Terminology and citations

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

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

The following terms are used as indicated:

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

OMEGAMON

Refers to any of the following products:

- IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS
- IBM Tivoli OMEGAMON XE for DB2 Performance Monitor on z/OS
- IBM DB2 Performance Expert for Multiplatforms and Workgroups
- IBM DB2 Buffer Pool Analyzer for z/OS

C, C++, and C language

Represent the C or C++ programming language.

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

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

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

RACF®

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

How to send your comments

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

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

- You can send comments from the Web. Visit the library Web site at:

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

This Web site has a an online reader comment form that you can use to send comments.

- You can also send comments by using the feedback link at the footer of each page in the Information Management Software for z/OS Solutions Information Center at <http://publib.boulder.ibm.com/infocenter/db2zhhelp>.

Chapter 1. The future is now

In a fast-paced world that never sleeps, the future is now. Are you ready? As the volume of customer transactions doubles, triples, and quadruples in shorter periods of time, can your database management system scale with those increasing demands? As the demands of your global business increase, the cycle time for developing and deploying new applications decreases: 18-month development cycles decrease to one month or less. What's more, you must take care not to interrupt support for existing applications while you deploy new applications. Can your IT organization and your current infrastructure deliver what you need when you need it?

Delivering customer value means the difference between winning and losing in the competitive environment of the twenty-first century. Seventy percent of customer value is driven by customer interactions. Ninety percent of customer interactions are tied to information, and 85 percent of data is unstructured. Leveraging data is the way to deliver customer value. Can your customers retrieve the right information quickly and easily—on demand? Can you retrieve the right information to maintain your competitive edge? Winning a new customer costs five times more than keeping a current customer. Are your current customers happy?

In the open world of your intranet and the Internet, is your data secure? You need a powerful on demand e-business infrastructure that lets you leverage your data, deliver customer value, and maintain your competitive edge without compromising the safety, integrity, and availability of your data. IBM delivers the power you need to leverage your data, deliver customer value, and keep your e-business competitive without compromising the safety, integrity, and availability of your data.

DB2 UDB for z/OS, IBM's flagship relational database management system, is ready to meet the challenge as the foundation on which your on demand e-business infrastructure is built. DB2 UDB for z/OS is robust, reliable, scalable, and open.

Tight integration with the zSeries® architecture and z/OS operating environment adds up to greater power, flexibility, and responsiveness for DB2:

In Version 8, DB2 UDB for z/OS takes advantage of the 64-bit virtual addressing support in z/OS and the z/Architecture™. With 64-bit virtual addressing, you get relief for virtual storage constraints, support for larger buffer pools, and more concurrent threads. These enhancements allow greater scalability of large DB2 subsystems or members of a DB2 data sharing group to meet the demands of increasing and unpredictable workloads. As a result, you keep up with the demands of your on demand e-business enterprise.

IBM continues to invest heavily in extending the technology leadership of the zSeries Parallel Sysplex® cluster architecture. Version 8 exploits new coupling facility commands that were developed specifically for DB2 to more efficiently write and cast out groups of pages with a single command instead of one command per page. These commands improve the efficiency of DB2 data sharing, especially for applications that update large amounts of data. In

addition, Version 8 implements a new locking protocol for data sharing to reduce locking overheads in the coupling facility and to provide greater availability.

Version 8 supports many of your evolving application needs while maintaining application availability:

Online schema changes, a major Version 8 enhancement, supports high resiliency and availability by letting you make some SQL data definition language changes without taking the database out of service. With online schema changes, you avoid the disruption and the extra time that is required to drop, re-create, or stop objects before making changes to your applications. For example, you can add a partition to an existing partitioned table space in a few seconds. You can also rotate partitions, and alter many attributes for tables, table spaces, and indexes.

The consistency of SQL across the DB2 family has improved substantially over the past few versions, even while new function has been added. Version 8 significantly enhances the portability and compatibility of your applications with other relational databases, including other members of the DB2 family, by drastically increasing the length of many SQL entities such as: SQL identifiers; character and hexadecimal string constants; table names, column names, view names, and alias names; predicates, and a subset of special registers.

Tight integration between DB2 UDB for z/OS, WebSphere[®], and Java[™] gives you the freedom to develop, test, and deploy your e-business applications when and where you need them. With Java, you save time and effort because you can code an application once and use it in many different environments.

Improvements to stored procedures include improved support for Unicode names and support for dynamic task management through the Workload Manager for superior scalability.

Version 8 also offers several autonomic functions that make restarting utilities and allocating space easier. With autonomic restart, you no longer need to add the RESTART parameters to a utility job—DB2 attempts to restart utilities that can be restarted online regardless of whether the RESTART keyword is specified. DB2 can also calculate secondary extents through a sliding scale algorithm. Autonomic space allocation improves performance, increases data availability, and limits the occurrence of outages that are caused by a lack of space.

DB2 supports your enterprise data needs more completely:

DB2 UDB for z/OS improves support for Java and Unicode with better flexibility, longer fields, and the ability to join Unicode data to EBCDIC data. SQL statements and literals can be in Unicode format, and much of the DB2 catalog data is stored in the Unicode format.

Support for materialized query tables lets you precalculate and store enterprise summary data; an automatic query rewrite function can then use the summary data to dramatically improve the performance of your complex decision-support queries. This support represents a major improvement for enterprise application vendors like SAP, PeopleSoft, Siebel, and many others.

Improved support for very large databases in Version 8 gives you more flexibility. You can now create more partitions, add new partitions with an ALTER statement, or rotate partitions. You can also define a clustering order that is different from the partitioning order to improve performance and availability. In addition, you can create a partition without an index to improve efficiency.

Integrated SQL support for XML in DB2 UDB for z/OS provides XML publishing functions that allow applications to generate XML data from relational data.

Read on to find out more about how the power of Version 8 of DB2 UDB for z/OS provides a solid foundation for your on demand e-business infrastructure.

Chapter 2. What's new in Version 8 of DB2 UDB for z/OS?

IBM delivers enhancements to DB2 UDB for z/OS that support your needs in the zSeries environment. Version 8 of DB2 UDB for z/OS delivers improvements in the following areas:

- "More flexibility with SQL"
- "Improved security" on page 8
- "Enhanced compatibility with the DB2 family" on page 10
- "Scalability and performance" on page 16
- "Reliability, availability, and serviceability" on page 23

More flexibility with SQL

Version 8 greatly expands support for SQL functions in online transaction processing environments through the following enhancements:

- "SELECT FROM INSERT statement"
- "Generation of unique sequential numbers for applications" on page 6
- "Ability to alter identity column attributes" on page 6
- "Dynamic scrollable cursors" on page 6
- "Scalar fullselects in SQL statements" on page 6
- "Integrated XML publishing functions in DB2" on page 6
- "Common table expressions in SQL statements" on page 7
- "Recursive SQL" on page 7
- "CURRENT PACKAGE PATH special register" on page 7
- "GET DIAGNOSTICS statement" on page 7
- "More than one DISTINCT keyword allowed in a single query" on page 12

A green, curved text graphic showing a SQL query: `SELECT EMPNO, LASTNAME, SALARY FROM DSN8610.EMP WHERE EDLEVEL > (SELECT AVG(EDLEVEL) FROM DSN8610.EMP);`

Figure 1. Version 8 of DB2 UDB for z/OS gives you more flexibility with SQL.

SELECT FROM INSERT statement

Recent releases of DB2 have provided enhancements (such as ROWID columns, identity columns, and triggers) in which DB2 or a trigger, rather than an application program, inserts data into DB2 tables. Before Version 8, you could not immediately determine the values that were inserted into DB2 tables by mechanisms other than application programs. Now, in Version 8, you can select values from rows that are being inserted into a DB2 table by specifying the SELECT FROM INSERT statement. The rows that are inserted into the target table produce a result table; the columns of the result table can be referenced in the SELECT list of the query. When you insert one or more new rows into a table, you can retrieve the following values from the result table:

- The value of an automatically generated column such as a ROWID column or an identity column

- Any column values that are the result of an expression
- Any default values for columns
- All values for an inserted row, without specifying individual column names
- All values that are inserted by a multiple-row INSERT operation
- Values that are changed by a BEFORE INSERT trigger

Generation of unique sequential numbers for applications

In prior releases and in Version 8, identity columns are used to generate sequential numbers; however, an identity column is a column of a table and is therefore associated with the table. Version 8 of DB2 UDB for z/OS introduces a new SQL data object, *sequence*, that provides recoverable, unique sequential numbers for applications. This data object also enhances the portability of your applications across the DB2 family and other vendor operating systems. In contrast to identity columns, sequences are stand-alone objects that applications can use to avoid concurrency and performance problems that can result when applications generate their own sequence numbers. After a sequence is defined, it can be concurrently accessed and incremented by many users, including multiple DB2 members in a data sharing group.

Ability to alter identity column attributes

Since identity columns were first introduced in Version 6, many users have asked for the ability to alter many of the attributes of identity columns. With Version 8, you can use the ALTER COLUMN clause of the ALTER TABLE statement to change all of the attributes of an identity column except the data type. The ability to alter attributes of identity columns eliminates the need to drop and re-create a table, which makes identity columns extremely flexible to use. For example, you can now alter the sequence attributes of existing identity columns, restart the column values from the new value, avoid minimum and maximum values, and control the order in which column values are generated.

Dynamic scrollable cursors

Version 7 provided the static scrollable cursor function in which scrolling is performed on a materialized global temporary table. Version 8 extends the scrollable cursor function by implementing a *dynamic scrollable cursor*. A dynamic scrollable cursor lets applications scroll directly on the base table while accessing the most current data, including newly inserted rows. The dynamic scrollable cursor function is particularly beneficial for large result sets that would otherwise need to be materialized. Dynamic scrolling is also supported by data-partitioned secondary indexes, index scans, and table space scans.

Scalar fullselects in SQL statements

Version 8 of DB2 UDB for z/OS adds more power to your queries and gives you more flexibility with your applications across the DB2 family. DB2 now supports scalar fullselects (with some restrictions) wherever expressions are allowed in SQL statements. Each scalar fullselect within an SQL statement returns a single row that consists of a single column.

Integrated XML publishing functions in DB2

In Version 8, DB2 UDB for z/OS enhances its leadership as an enterprise database server by providing a set of SQL built-in functions that allow applications to generate XML data from relational data with high performance. The XML

publishing functions can reduce application development efforts in generating XML data for data integration, information exchange, and Web services. With XML publishing functions, DB2 can:

- Generate XML elements with optional attributes from columns and expressions
- Generate XML data with hierarchical structures, through grouping and concatenation, for data that has parent-child relationships

Common table expressions in SQL statements

Version 8 provides improved usability and consistency across the DB2 family through support for common table expressions in SQL statements. A common table expression is like a temporary view that is defined and used for the duration of an SQL statement. You can reference each common table expression many times in an SQL statement; all references to a common table expression share the same result table. In contrast, regular views or nested table expressions are derived each time that they are referenced. Common table expressions can also improve performance in some cases because values are computed once rather than several times.

Recursive SQL

In Version 8, you can use common table expressions to create recursive SQL. If a fullselect of a common table expression contains a reference to itself in a FROM clause, the common table expression is known as a *recursive common table expression*. Common table expressions and recursive SQL improve usability and consistency within the DB2 UDB family. In some cases, common table expressions and recursive SQL can be used to improve performance because values are derived once rather than several times. Queries that use recursion are useful in applications like bill-of-materials applications, network planning applications, and reservation systems.

CURRENT PACKAGE PATH special register

Support for a new special register, CURRENT PACKAGE PATH, reduces network traffic, simplifies application coding, and improves processing time and elapsed time for stored procedures, user-defined functions, Java programs that use SQLJ, and applications that use DRDA from a z/OS requester. Many installations use more than one collection for packages. In prior releases, applications that do not use plans must issue the SET CURRENT PACKAGESET statement each time a package from a different collection is used.

With support for the CURRENT PACKAGE PATH special register in Version 8, an application programmer can specify a list of package collections in one SET CURRENT PACKAGE PATH statement. The new SET CURRENT PACKAGE PATH special register also lets you implement a nested procedure or a user-defined function, regardless of your run-time environment, and lets you specify multiple collections.

GET DIAGNOSTICS statement

The GET DIAGNOSTICS statement returns diagnostic information about the previous SQL statement that was executed, and is consistent with the ANSI/ISO Core Level SQL Standard for 1999. The GET DIAGNOSTICS statement in Version 8 is more robust and less restrictive than the SQLCA. You can use the GET DIAGNOSTICS statement to request information about:

- The entire SQL statement. In addition, the GET DIAGNOSTICS statement can return longer names.

- Conditions, including multiple conditions for multiple-row statements, and the error message that is associated with an individual error. GET DIAGNOSTICS supports SQL error message tokens that are greater than 70 bytes.
- Connections, if the SQL statement was a CONNECT statement.

You can issue the GET DIAGNOSTICS statement from an embedded application and from within an SQL procedure.

Compare null values with the DISTINCT predicate

The new DISTINCT predicate lets you compare null values and simplifies the SQL that you need to write when you need to find values that might be null. Two forms of the DISTINCT predicate are:

IS DISTINCT FROM

Creates an expression in which both values are not equal or one value is null.

IS NOT DISTINCT FROM

Creates an expression in which one value is equal to another value, or both values are null.

Improved security

Version 8 provides new options for e-business and high security with multilevel security and row level security. Together, these improvements let you identify system users more easily and significantly increase the granularity of your security authorizations. These improvements result in additional flexibility for applications and SQL. Other new features, such as encryption, also improve security. The following sections provide more details about security improvements in Version 8:

- “New built-in functions for data encryption and decryption”
- “New data encryption tool” on page 9
- “Multilevel security with row-level granularity” on page 9
- “Easier identification of system users” on page 9
- “Session variables” on page 10
- “Improved encrypted security in distributed computing environments” on page 10

New built-in functions for data encryption and decryption

Version 8 of DB2 UDB for z/OS provides new built-in functions for data encryption and decryption that let you protect valuable data as it is stored in or retrieved from a DB2 subsystem. The ENCRYPT function lets you encrypt and store data in columns of DB2 tables. You can copy, restore, or move encrypted data between DB2 subsystems.

DB2 gives you the flexibility to encrypt all values in a column of data with the same (or *common*) password or to allow many different passwords within a column. For example, you can set up a common password that lets users access a specific view of the data. Or you can let individual users create their own passwords when they set up an account with their credit card numbers and related information on an e-business Web site.

Additional built-in functions provide support for generating unique sequential values in a table, decrypting encrypted data, and setting up password protection.

New data encryption tool

IBM Data Encryption for IMS and DB2 Databases, one of the IBM DB2 and IMS Tools products, is a single tool that you can use to protect sensitive IMS and DB2 UDB for z/OS data. In DB2, data encryption and decryption is implemented through the standard EDITPROC exit routine. The exit code uses the zSeries and S/390[®] Crypto Hardware to encrypt data for storage and decrypt data for application use. This tool can help you save the time and effort that is required to write and maintain your own encryption software. For additional information about IBM Data Encryption for IMS and DB2 Databases, see www.ibm.com/software/data/db2imstools/

Multilevel security with row-level granularity

Multilevel security is a security policy that lets you classify data and users based on a system of hierarchical security levels that is combined with a system of nonhierarchical security categories. The goals of multilevel security are twofold: To prevent individuals from accessing information that is classified at a level that is higher than their authorization allows, and to prevent individuals from declassifying information.

Version 8 of DB2 UDB for z/OS supports multilevel security with row-level granularity, which lets you restrict individual user access to a specific set of rows in a table. Multilevel security with row-level granularity offers several advantages over current authorization techniques:

- Security enforcement is mandatory and automatic; a user is checked at run time. This technique complements existing discretionary checks.
- You can perform security checks that are difficult to express through traditional SQL views or queries.
- Multilevel security does not rely on special views or database variables to provide row-level security control.
- Security controls are consistent and integrated across the system so that you can avoid defining users, objects, access, and security labels more than once. Access to files, database, printers, terminals, and other resources can have a single security control point.

Easier identification of system users

Prior to Version 8, if you have a large server, such as WebSphere Application Server, connected to DB2, all the threads for a connection show a single identity (the server). Now, Version 8 provides four new special registers that you can use in your applications to more easily identify system users.

CLIENT_ACCTNG special register

The CLIENT_ACCTNG special register contains the current value of the accounting string from the client information that is specified for a connection. Now you can get the current value of the accounting string that is used in a specific connection. For example:

```
SET :ACCT_STRING = CLIENT_ACCTNG
```

CLIENT_APPLNAME special register

The CLIENT_APPLNAME special register contains the value of the application name from the client information that is specified for a connection. Now you can select which departments can use the application that is used in a specific connection. For example:

```
SELECT DEPT
  FROM DEPT_APPL_MAP
 WHERE APPL_NAME = CLIENT_APPLNAME
```

CLIENT_USERID special register

The CLIENT_USERID special register contains the value of the client user identifier from the client information that is specified for a connection. Now you can find out in which department a current client user ID works. For example:

```
SELECT DEPT
  FROM DEPT_USERID_MAP
 WHERE USER_ID = CLIENT_USERID
```

CLIENT_WRKSTNNAME special register

The CLIENT_WRKSTNNAME special register contains the value of the workstation name from the client information that is specified for a connection. Now you can get the workstation name that is used for a particular connection. For example:

```
SET :WS_NAME = CLIENT_WRKSTNNAME
```

Session variables

Support for session variables in Version 8 gives you another means by which to provide information to applications. DB2 sets some session variables, and you can set other session variables in the connection and signon exit routines. A new built-in function, GETVARIABLE, retrieves the values of a session variable. You can use this function to enforce security policies in views, triggers, stored procedures, and constraints. Application programmers and SQL users can now access the information that is set by DB2: Plan names, package names, DB2 version identifiers, security labels, and system CCSIDs. If you require more general, flexible controls for primary security, you can use session variable information to complement other security mechanisms.

Improved encrypted security in distributed computing environments

New Distributed Relational Database Architecture™ (DRDA®) security options provide the following data security improvements in distributed computing environments:

- DB2 UDB for z/OS servers can provide secure, high-speed data encryption and decryption.
- DB2 UDB for z/OS requesters now have the option of encrypting user IDs and optionally, passwords when they connect to remote servers. Requesters can also encrypt security-sensitive data when they communicate with servers, so that the data is secure when it travels over the network.

Enhanced compatibility with the DB2 family

The drive toward family compatibility continues in Version 8 with:

- “Extended limits for names and SQL statements” on page 11
- “Significant support for Unicode and long names in the DB2 catalog” on page 12
- “Network computing enhancements” on page 13
- “IBM z/OS Application Connectivity to DB2 for z/OS feature” on page 15
- “ODBC expands support for encoding schemes” on page 16

Most of the enhancements that make SQL more flexible, as discussed in “More flexibility with SQL” on page 5, also improve compatibility across the DB2 family.

The best reference for developing applications that are portable across the DB2 family is *IBM DB2 Universal Database SQL Reference for Cross-Platform Development*, available at www.ibm.com/software/db2zos/library.html

This book is updated as changes in SQL language elements are implemented across members of the DB2 family.

Extended limits for names and SQL statements

Version 8 of DB2 UDB for z/OS takes a giant leap over the current limits for column names, for SQL statements, for index keys and predicates, and for the number of tables that can be joined in a single FROM clause with the following enhancements:

Longer column names

In Version 8, the maximum length for table and view names is extended from 18 bytes to 128 bytes. The maximum length of column names is changed from 18 bytes to 30 bytes. Names for many other objects, such as indexes, statements, schemas, procedures, and triggers are also extended to 128 bytes.

Longer and more complex SQL statements

SQL statements can now be up to 2 MB in length. A number of the Version 8 capabilities stretch the limit on the size of an SQL statement. Long names and support for up to 4096 partitions require more space. An SQL procedure must be stated completely within a single SQL statement. Other changes in DB2 allow larger structures and, therefore, larger statements. SQL statements that are too large or too complex become very rare in the Version 8 environment.

Extensions to SQL procedure statements

In Version 8, the 2-MB extension to the length of an SQL statement also applies to the CREATE PROCEDURE statement. Specifically, the length of an individual SQL procedure statement, which consists of SQL control statements and SQL statements in the procedure body, is extended to 2 MB. As a result, if you specify an SQL control statement as the procedure body, you can include multiple SQL procedure statements within that control statement, each of which is now extended to 2 MB. This enhancement significantly increases the power and flexibility of SQL procedures.

Longer index keys and predicates

The maximum length of an index key is increased from 255 bytes to 2000 bytes. Similarly, the maximum length for a predicate operand is increased from 255 bytes to 32 704 bytes, which is the maximum length of a VARCHAR column.

Greater number of tables joined in a single FROM clause

Prior releases of DB2 let you join up to 15 tables in a single FROM clause. Many users need to run queries that join more than 15 tables. In some cases, users need to join as many as 80 tables; the trend is moving toward joining even more tables. Version 8 continues to meet user needs by offering support for joining up to 225 tables in a single FROM clause.

Fewer restrictions for column functions

The argument of a column function is a set of like values that is derived from an expression. Prior to Version 8 of DB2, the expression for the argument was required to include a reference to a column (which was referred to as a *column function*). In Version 8, you no longer need to specify a column name in the expression. Because a column reference is no longer required, column functions are now called *aggregate functions*.

Qualified columns in the INSERT statement

In prior releases of DB2, you cannot qualify the names of columns when you insert data into a column. In Version 8, you can use qualified column names in an INSERT statement just like you can in an UPDATE statement.

ORDER BY clause for the SELECT INTO statement

The SELECT INTO statement must produce a result that contains a single row. Prior to Version 8, you could specify the FETCH FIRST 1 ROW clause to ensure that only a single row was returned if the result set of the query could result in more than one row. However, you could not specify the ORDER BY clause to affect which row was returned. With Version 8, you can now specify ORDER BY. When you use both the FETCH FIRST 1 ROW and ORDER BY clauses, the result set is ordered first, and then the first row is returned.

Expressions in the GROUP BY clause

To enable greater portability of applications, Version 8 of DB2 UDB for z/OS lets you specify the same expressions in the GROUP BY clause that you can specify in HAVING, SELECT, and ORDER BY clauses. Now you can have family consistency without rewriting your SQL statements.

More than one DISTINCT keyword allowed in a single query

Version 8 enhancements to SQL support give you the flexibility to use more than one DISTINCT keyword in a single query. As a result, you no longer need to write multiple queries to retrieve multiple distinct column values.

Additional input format for timestamp strings

In addition to using a dash to separate the date portion and the time portion of a timestamp string, you can now use a blank as the separator. The ODBC and JDBC string representations of a timestamp use the format in which the blank is the separator.

Explicitly defined ROWID columns are not required for large objects

In Version 8, you no longer need to explicitly define a ROWID column when you define a large object (LOB) column. If a ROWID column does not exist when you define a LOB column with either the ALTER TABLE or CREATE TABLE statement, DB2 implicitly generates a ROWID column.

Descriptions for plans and packages in the DB2 catalog

You can now provide descriptions for plans and packages in the DB2 catalog. Support for comments for plans and packages simplifies documenting and tracking your objects and increases compatibility within the DB2 UDB family.

Implicit drop of declared global temporary tables at commit

In Version 8, you can specify that DB2 is to implicitly drop declared global temporary tables at a commit operation. The new ON COMMIT DROP TABLE clause of the DECLARE GLOBAL TEMPORARY TABLE statement lets you drop the declared global temporary table at commit if no open cursors on the table are defined as WITH HOLD. This enhancement is particularly important for distributed applications and stored procedures because cleanup can occur when cursors are closed.

Significant support for Unicode and long names in the DB2 catalog

Architectural changes in Version 8 expand the DB2 catalog for long names and Unicode.

Unicode

With Version 8 comes significant support for Unicode, which means that you can manage data from around the world. DB2 now converts any SQL statement to Unicode before parsing; as a result, all characters parse correctly. DB2 also supports hexadecimal string constants.

Long names

The following enhancements for long names in Version 8 make DB2 UDB for z/OS compatible with other members of the DB2 family:

- Longer string constants (up to 32 704 bytes)
- Longer index keys (up to 2000 bytes)
- Longer predicates (up to 32 704 bytes)
- Longer object names (up to 30 characters for column names and up to 128 characters for most other SQL objects)

Unicode, EBCDIC, and ASCII columns in the same SQL statement

Prior to Version 7, DB2 supported a limited set of coded character set identifiers (CCSIDs) to store data in EBCDIC and ASCII encoding schemes. Version 7 of DB2 for z/OS and OS/390 introduced the Unicode encoding scheme to address the problems that are encountered when users who live in different geographies and who speak many different languages interact with the same DB2 server. The Unicode encoding scheme represents the characters of many different geographies and languages. With Version 7 support for the Unicode encoding scheme, you cannot reference table objects that are defined with different encoding schemes in the same SQL statement.

Version 8 further expands DB2 UDB for z/OS support of multiple encoding schemes by letting you reference tables or table functions with different CCSIDs in the same SQL statement. Support for multiple CCSIDs gives you the flexibility to join tables that have Unicode, EBCDIC, and ASCII columns.

Network computing enhancements

Version 8 of DB2 UDB for z/OS provides the following enhancements for network computing:

- “Increased portability of applications through consistent access to DB2 family servers”
- “Improved performance for remote queries” on page 14
- “Improvements in connectivity” on page 15

Increased portability of applications through consistent access to DB2 family servers

In prior releases, differences in access paths for applications that run on DB2 UDB for Linux, UNIX[®] and Windows[®] and on DB2 UDB for z/OS required duplication of effort in developing and testing applications for the different run-time environments. In addition, accessing a DB2 UDB for Linux, UNIX and Windows server and a DB2 UDB for z/OS server required different database connection protocols. Each connection protocol, in turn, defined a different set of methods to implement the same functions.

Enhancements in Version 8 remove roadblocks to performance and DB2 family compatibility by providing support for a common client and standardizing database connection protocols based on the Open Group Technical Standard DRDA Version 3. Three components comprise the common client: A C common client for ODBC, a Java common client for SQLJ and JDBC, and an administrative client.

Together, these components provide consistent access to servers across DB2 environments through a common run-time environment that has a single access path for all applications. (See Figure 2.) As a result, new function and new applications can go into production more quickly because you can write an application once and use it in any of the DB2 common client environments.

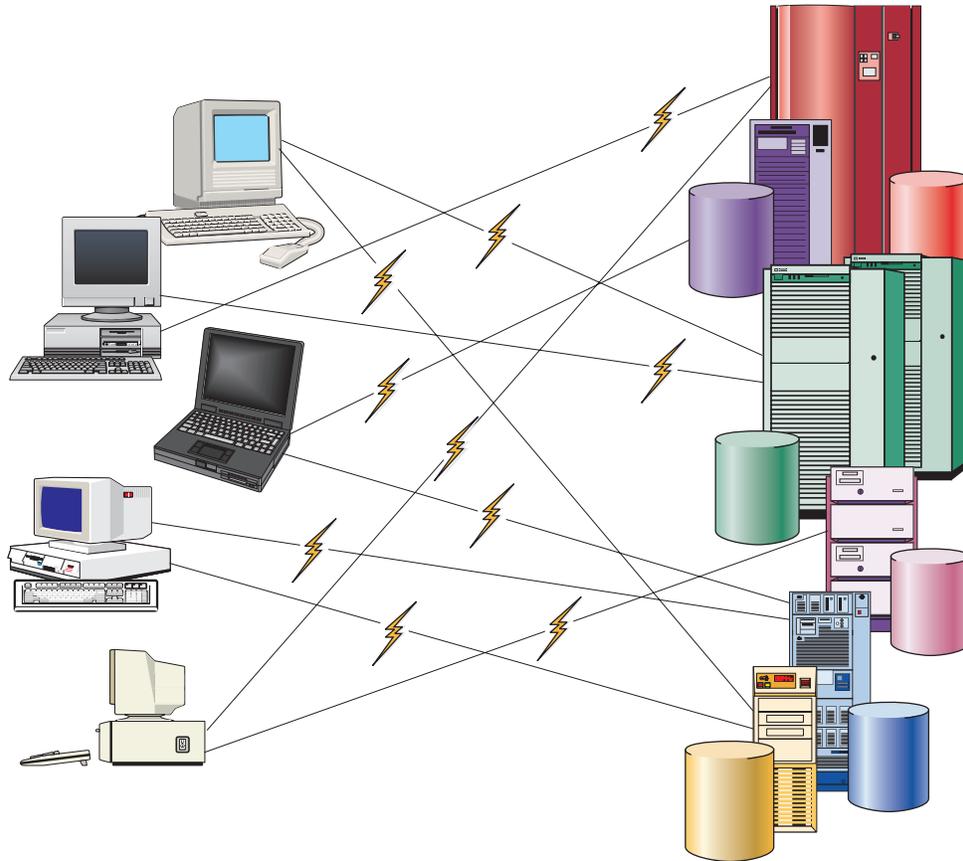


Figure 2. The DB2 common clients offer increased portability of your applications through consistent access to DB2 family servers.

Improved performance for remote queries

Version 8 provides the following new function to improve performance for remote queries:

- Version 8 provides a new server interface to boost the performance of processing remote fetches.
- Multiple-row fetch and multiple-row insert functions improve the DRDA interface to DB2 and provide a more efficient approach for generating query blocks. DB2 builds the query directly into the communication buffer in a single request, which minimizes the number network operations that are required to generate a single query block.
- Improvements to DRDA support in DB2 improve the performance of read-only, single-row cursors.
- The SQL cancel function allows a JDBC or CLI application to cancel long-running requests on a DB2 UDB for z/OS server.

Improvements in connectivity

As a result of support for Version 3 of Open Group Technical Standard DRDA, the following connectivity improvements in Version 8 of DB2 UDB for z/OS are available:

- A DB2 database for Linux, UNIX, and Windows is known in the network by its database name. Applications use the database name to connect to an instance of a DB2 UDB for Linux, UNIX and Windows database. When DB2 is deployed to a large number of servers in a Linux, UNIX, or Windows environment, the database is commonly deployed with the same name at all locations. A database administrator can now specify (in the new DBALIAS column in the table SYSIBM.LOCATIONS) multiple locations for a DB2 UDB for Linux, UNIX and Windows database that is deployed in multiple locations. As a result, a DB2 UDB for z/OS requester can now access multiple DB2 databases in the Linux, UNIX, and Windows environments that have the same name but different network addresses.
- A DB2 server is known in a network by its location name. Applications use the location name to identify an instance of a DB2 subsystem or a group of DB2 subsystems that share data. When you migrate two or more DB2 subsystems to a single data sharing group, you must consolidate multiple locations into a single location (migrating several subsystems to a single data sharing group is a complex task that might require other tools). After the locations are consolidated, you must change all applications that use the old location name to access the location name of the new data sharing group. When a large number of remote applications are deployed across a network, changing each application to use the new location name simultaneously is difficult. To support migration from multiple locations to a single location, DB2 lets you define eight location alias names for a DB2 subsystem or for a group of data-sharing DB2 subsystems. A location alias is another name that a requester can use to access a DB2 subsystem.
- You can also use a location alias with a TCP/IP port number to allow connections to a subset of data sharing members from DRDA requesters that connect to DB2 UDB for z/OS through TCP/IP.
- A database administrator can now override the automatic TCP/IP workload balancing function in a data sharing environment by setting up rows in the new SYSIBM.IPLIST table at a requester in conjunction with defining location aliases at the server. With the new SYSIBM.IPLIST table, a database administrator can define a specific member or a subset of members in a data sharing group. With this new service, applications can route requests by using a name that is different from the group location name.
- Version 8 also adds DRDA XA protocol support, which enables distributed transactions that implement the Java 2 Platform, Enterprise Edition (J2EE) Java Transaction Service (JTS), and Java Transaction API (JTA) specifications.

IBM z/OS Application Connectivity to DB2 for z/OS feature

z/OS Application Connectivity to DB2 for z/OS is a no-charge, optional feature of DB2 UDB for z/OS. This feature consists of a component known as the DB2 Universal Database Driver for z/OS, Java edition. This pure Java, type 4 JDBC driver is designed to deliver high performance and scalable remote connectivity for Java-based enterprise applications on z/OS to a remote DB2 for z/OS database server. The driver:

- Supports JDBC 2.0 and 3.0 specifications and Java Development Kit, Version 1.4 to deliver the maximum flexibility and performance that is required for enterprise applications

- Delivers robust connectivity to DB2 for z/OS and the WebSphere Application Server for z/OS
- Supports distributed transactions
- Allows custom Java applications that do not require an application server to run in a remote partition and connect to DB2 for z/OS

The DB2 Universal JDBC Driver is a single driver that includes JDBC type 2 and JDBC type 4 behavior and SQLJ support. When an application loads the DB2 Universal JDBC Driver, a single driver instance is loaded for type 2 and type 4 implementations. The application can make type 2 and type 4 connections by using this single driver instance. The DB2 Universal JDBC Driver supports the following JDBC and SQLJ functions:

- Most of the methods that are described in the JDBC 1.2 and JDBC 2.0 specifications and some of the methods that are described in the JDBC 3.0 specifications
- Connection pooling
- Global transactions that run on WebSphere Application Server, Version 5.0 and later
- Distributed transaction support that implements the Java 2 Platform, Enterprise Edition (J2EE) Java Transaction Service (JTS), and Java Transaction API (JTA) specifications

ODBC expands support for encoding schemes

ODBC now supports Unicode formats UTF-8 and UCS-2. In addition, a new ODBC initialization file keyword, CURRENTAPPENSCH, lets you specify the encoding scheme that you want the ODBC driver to use for input and output of host variable data, SQL statements, and all character string arguments of the ODBC application programming interfaces. You can specify one of the following encoding schemes: Unicode, EBCDIC, or ASCII.

Scalability and performance

In Version 8, DB2 breaks through limits and sets new heights for scalability and performance with:

- “64-bit virtual storage” on page 17
- “Materialized query tables” on page 18
- “Ability to use an index in more situations” on page 18
- “Additional distribution statistics for improved optimization” on page 19
- “Improved optimization for dynamic SQL” on page 19
- “Improved trigger performance” on page 19
- “More parallelism for sort operations” on page 20
- “Performance enhancements for star join qualified queries” on page 20
- “Multiple fetches and inserts allowed within a single SQL statement” on page 20
- “4096 partitions in a partitioned table space” on page 20
- “Greater resource control with stored procedures and user-defined functions” on page 22
- “Reduced overhead costs for data sharing workloads” on page 22
- “Reduced lock propagation in the coupling facility” on page 22
- “Improved control for accounting aggregation” on page 22
- “Improved package-level accounting” on page 23

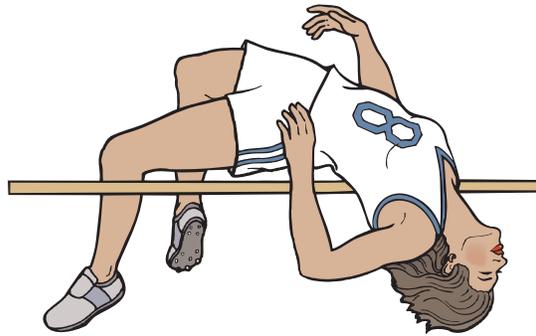


Figure 3. Version 8 of DB2 UDB for z/OS breaks through limits and sets new heights for scalability and performance.

64-bit virtual storage

Version 8 of DB2 UDB for z/OS, through exclusive integration with the IBM zSeries 800, 900, or the equivalent, now supports 64-bit virtual storage. The zSeries 64-bit architecture allows DB2 UDB for z/OS to move various storage areas above the 2^{31} -byte (2-GB) bar. The single large address space of up to 2^{64} bytes (16 exabytes) in Version 8 is 8 billion times larger than the address space that is available in Version 7. How big is 8 billion times larger? It is the difference between 5 centimeters and the distance from the earth to the moon. (See Figure 4.) Or, consider that if you start now with 2-GB of virtual storage and double it every year, you will reach the limit of 16 exabytes in 33 years. The immense size of the address space in Version 8, no matter how you visualize it, improves scalability and availability and gives you more flexibility to manage virtual storage.

How big *is* 8 billion times larger?

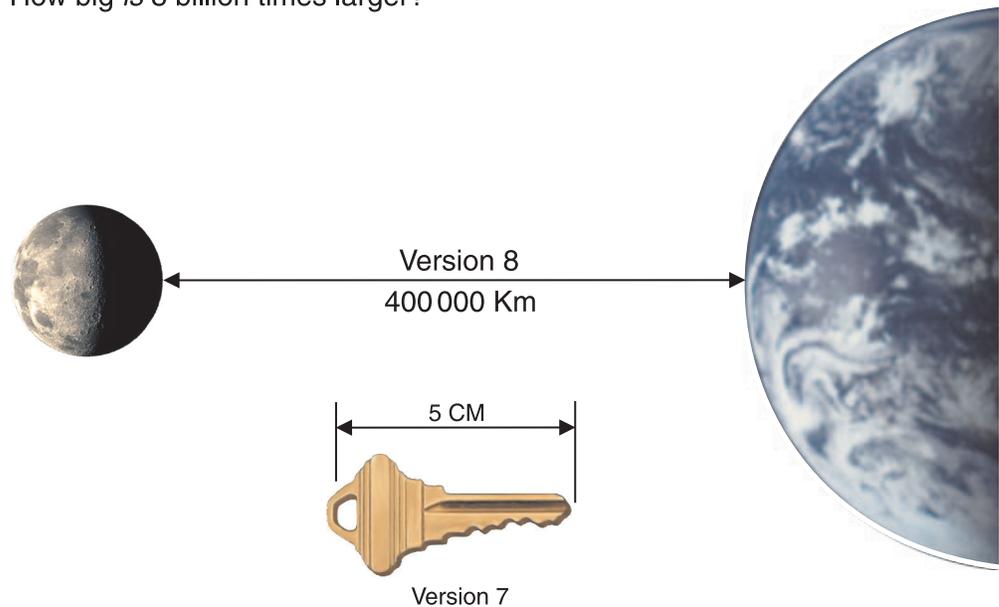


Figure 4. Relative size comparison of 31-bit virtual storage on OS/390 to 64-bit virtual storage on z/OS

Materialized query tables

Decision-support queries typically operate on large amounts of data (1 to 10 terabytes) to perform multiple joins and complex aggregation. To reduce the elapsed time of decision-support queries, Version 8 of DB2 UDB for z/OS supports materialized query tables. A materialized query table contains materialized data that is derived from one or more source tables that are specified by a fullselect in an SQL expression. (See Figure 5.) DB2 can then use the materialized data to answer a query more efficiently. A new clause on the CREATE TABLE statement lets you control whether a materialized query table is to be used automatically to answer queries. DB2 also lets you specify, within a query, whether you want DB2 to take advantage of available materialized query tables.

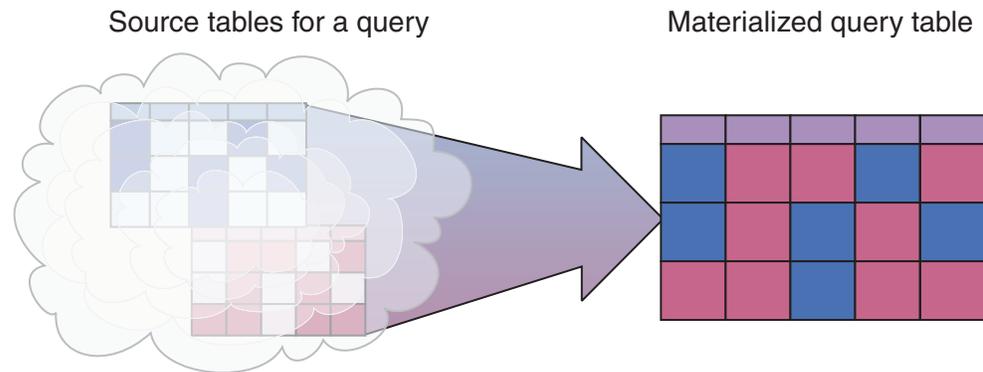


Figure 5. An illustration of a materialized query table that is derived from two source tables

Ability to use an index in more situations

Version 8 of DB2 UDB for z/OS gives you the flexibility to use an index in more situations for improved scalability and better performance. Improvements include:

- “Capability to index predicates that have mismatched data types”
- “Stored variable-length index keys”
- “Backward index scans for avoiding sort operations” on page 19

Capability to index predicates that have mismatched data types

In Version 8, you can now index many predicates that have mismatched data types; as a result, query performance improves. You can join tables on columns that have different data types and lengths, or you can provide a search value with a data type or length that does not match the definition of a column. For example, the C and C++ programming languages do not support the decimal data type, so programs that are written in these languages often use the floating-point data type for predicates on decimal columns. With Version 8 support for mismatched data types and lengths, predicates like those in the preceding example can now be stage 1 predicates.

Stored variable-length index keys

In Version 7, an index-only access path cannot be used for short VARCHAR host variables. In addition, varying-length columns that consist of VARCHAR and VARGRAPHIC data are varying-length columns in tables, but they are padded to their maximum length in index keys. Indexes that are padded require additional storage.

As Figure 6 on page 19 shows, Version 8, support for true varying-length keys in the index allows index-only access for indexes that have varying-length index keys and reduces storage requirements. In most cases, index keys in which

varying-length columns are not padded require less storage because only data is stored in the index. Version 8 also gives you the flexibility to create or alter indexes so that they have varying-length columns in the keys, and you can control whether DB2 pads those columns in the index.

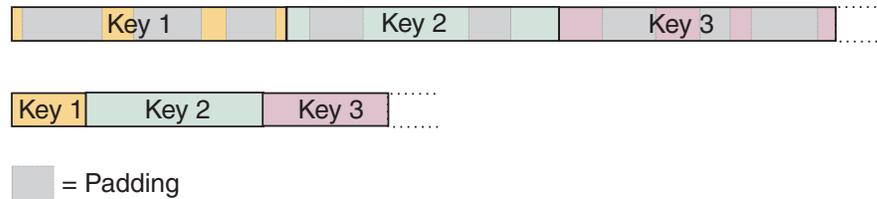


Figure 6. An illustration of true varying-length index keys in Version 8 of DB2 UDB for z/OS

Backward index scans for avoiding sort operations

Version 8 provides the capability for backward index scans, which lets DB2 avoid a sort. A sort is avoided when the DB2 optimizer determines that an ORDER BY clause can be satisfied by traversing an existing index in a backward direction. In prior releases, database designers would sometimes create a descending index for the sole purpose of avoiding a sort. Such descending indexes can now be deleted to improve performance because of the savings in index maintenance and reduced disk usage.

Additional distribution statistics for improved optimization

To run efficiently, data warehousing, data mining, and ad hoc query applications need statistics on columns that are in predicates, regardless of whether they are leading columns of an index. In addition, distribution statistics on non-leading index columns or non-indexed columns let DB2 make better access path decisions when data is asymmetrically distributed.

In Version 8, you can use the RUNSTATS utility to collect the following additional statistics:

- Frequency distributions for non-indexed columns or groups of columns
- Cardinality values for groups of non-indexed columns
- Least-frequently occurring values, most-frequently occurring values, or both, for any group of columns

Improved optimization for dynamic SQL

Several enhancements improve optimization for dynamic SQL:

- A new bind option, REOPT(ONCE), allows DB2 to determine and store only once at run time the access path for any dynamic SQL statement that contains variable values. DB2 uses the first set of input variables to determine the access path. The REOPT(ONCE) option can increase efficiency for dynamic SQL statements that run multiple times because DB2 does not reoptimize the access path each time a statement runs.
- A new clause of the EXPLAIN statement lets you obtain information about statements that have been stored in the dynamic statement cache. As a result, you can examine the current access path.

Improved trigger performance

In Version 8, DB2 requires fewer work files for processing conditional triggers. As a result, trigger performance improves.

More parallelism for sort operations

In Version 8, DB2 performs more parallel processing of some sort operations that are used in join processing. To ensure that parallel sort operations are cost-effective, DB2 uses a cost model to determine whether to perform a parallel sort.

Performance enhancements for star join qualified queries

The performance of a star join is critical to data warehousing applications in which the main database design is a star schema. The star join implementation in DB2 UDB for z/OS must handle a large number of work files. Because work files prior to Version 8 do not have indexes, sort-merge joins tend to be selected. As a result, the cost of sorting can be great in both time and the amount of space required.

Version 8 provides enhancements that improve the optimization and execution of a star join qualified query. Now, sparse indexes are supported on star join work files. As a result, the DB2 optimizer selects the access path based on the estimated costs of the access plans, which can boost query performance and can reduce the cost of sorting a larger number of work files.

Multiple fetches and inserts allowed within a single SQL statement

You can enhance the performance of your application programs by using multiple-row FETCH and INSERT statements to request that DB2 send multiple rows of data, at one time, to and from the database. Using these multiple-row statements in local applications results in fewer accesses of the database. Using these multiple-row statements in distributed applications results in fewer network operations and a significant improvement in performance. Figure 7 illustrates the difference between a series of single fetches and a single, multiple-row fetch operation.

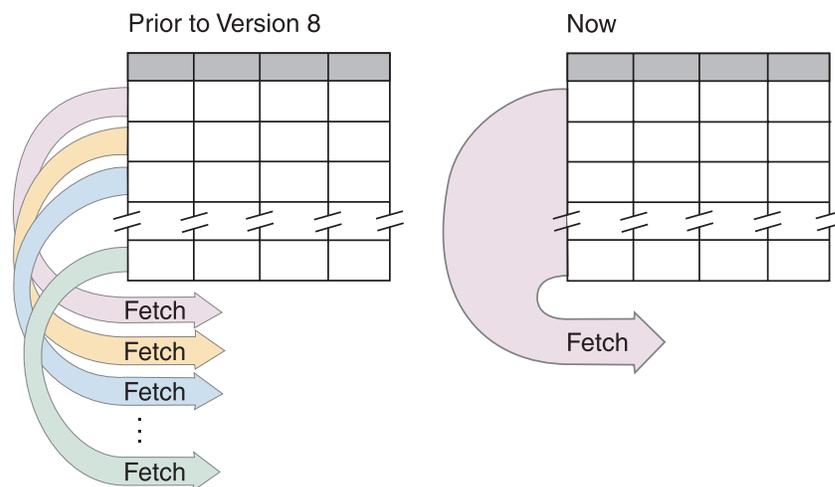


Figure 7. Before Version 8, you needed to use a series of single-row fetches to return many rows of data. Now, in Version 8, you can use multiple fetches within a single SQL statement to accomplish the same task more efficiently.

4096 partitions in a partitioned table space

As Figure 8 on page 21 shows, the maximum number of partitions in a partitioned table space is increased from 254 to 4096 partitions in Version 8. If you consider

that use of 254 partitions allows you to use one partition per day for 8 months, an increase to 4096 partitions extends those 8 months to 11 years.

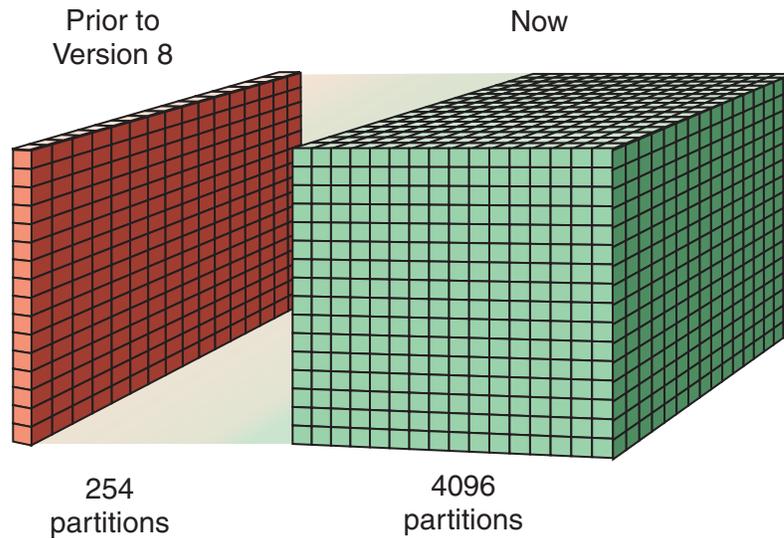


Figure 8. Comparison of 254 partitions to 4096 partitions in a partitioned table space

As a result of the increased number of partitions in a partitioned table space, the maximum size of a partitioned table is increased from 16 terabytes to 128 terabytes, as shown in Figure 9.

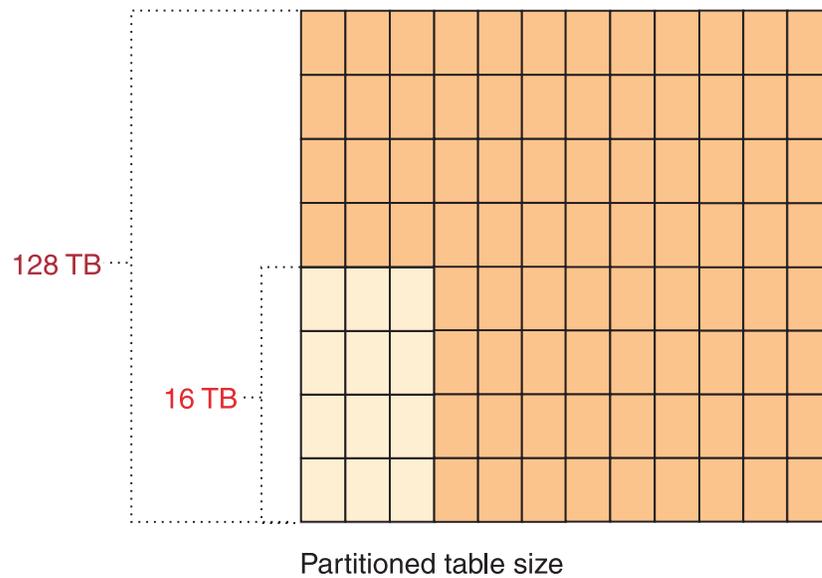


Figure 9. Comparison of a 16-terabyte partitioned table to a 128-terabyte partitioned table

The ability to have more partitions increases the granularity of your data and decreases the size of your data sets. For example, if one of your applications (prior to Version 8) stores daily data in separate partitions for one year, you would need up to 366 partitions. If you keep that data for five years, you would need 1830 partitions. To work around the 254-partition limit, you would need to either combine multiple days or weeks into partitions, or use simple table spaces and lose the benefit of parallel processing for queries. With Version 8, you reap the advantages of parallel processing for queries and utilities by using partitioned

table spaces: You have the flexibility to start with a minimum number of partitions and add more partitions as your needs change.

Greater resource control with stored procedures and user-defined functions

In Version 8, DB2 UDB for z/OS gives you greater control over resource utilization. You can now specify for each stored procedure or user-defined function the maximum number of failures (program exceptions or abnormal termination, for example) that are allowed before DB2 stops the routine. By specifying the most appropriate value for an individual routine, you can let some routines continue to be invoked for development and debugging, and stop other routines for maintenance before they cause problems in a production environment. Additional enhancements take advantage of z/OS Workload Manager functions that let System Resource Manager and Workload Manager determine appropriate resource utilization and recommend changes in the number of tasks that operate in a stored procedure address space. The stored procedure manager then adds or deletes tasks according to recommendations from Workload Manager.

Reduced overhead costs for data sharing workloads

Version 8 of DB2 UDB for z/OS provides two new batch processes that reduce the amount of traffic to and from the coupling facility when you run Version 1 Release 4 of z/OS and level 12 of the coupling facility. Two new commands let you write and register multiple pages to a group buffer pool, and read multiple pages from a group buffer pool for castout processing. You can expect the greatest performance benefits for data sharing workloads that update a large number of pages that belong to group-buffer-pool-dependent objects.

Reduced lock propagation in the coupling facility

Version 8 improves the performance of plans and packages that are bound with `RELEASE(COMMIT)` and increases the availability of your data because retained parent L-locks no longer lock an entire table or a table space when a DB2 member fails. You can now grant parent L-locks locally without invoking global contention processing. As a result, locking overhead that is caused by false contention is reduced.

Improved control for accounting aggregation

New e-business workloads that use the Recoverable Resource Manager Services attachment facility (RRSAF) or the distributed data facility (DDF) can generate enormous volumes of accounting records. To help reduce the volume of accounting records, Version 8 provides a new installation option (a subsystem parameter) that lets you accumulate data for RRSAF and DDF threads based on the following criteria:

- The user ID of the end user
- The transaction name of the end user
- The workstation name of the end user

Of course, some activities (detailed performance monitoring, for example) require detailed accounting data for RRSAF and DDF threads. With Version 8, you can dynamically alter the subsystem parameter to activate or deactivate data accumulation, which gives you the flexibility to meet your performance monitoring requirements.

Improved package-level accounting

New e-business workloads often use packages to issue SQL statements. With Version 8, package-level accounting contains additional detailed performance metrics that let you easily determine which packages were used to issue the SQL statements in your application workload. These metrics greatly simplify performance analysis activities for stored procedure applications, SQLJ applications, and DDF applications.

Reliability, availability, and serviceability

To keep your business competitive, the foundation of your e-business infrastructure must be reliable, available, and serviceable. Version 8 of DB2 UDB for z/OS continues to provide a strong foundation through the following enhancements:

- “Greater availability and flexibility with online schema evolution”
- “Greater data availability with data-partitioned secondary indexes” on page 25
- “More flexibility and faster recovery with system-level point-in-time recovery” on page 25
- “Improved utility functions” on page 26
- “Enhanced LPL recovery processing” on page 27
- “Change more parameters online without recycling DB2” on page 28
- “Increased maximum number of active and archive log data sets” on page 28
- “Autonomic space allocation” on page 28

Greater availability and flexibility with online schema evolution

In prior releases, your data and applications were unavailable during operations to alter schema definitions of table, table space, and index attributes, or to add, rotate, or rebalance partitions. Now, in Version 8, you can change schema definitions online for some table, table space, and index attributes or add, rotate, or rebalance partitions without losing availability of your data or applications. (See Figure 10 on page 24.) For example, you can change column types and lengths, add columns to an index, add, rotate, or rebalance partitions. Related enhancements provide more flexibility to change the clustering and partitioning of the data in a table.

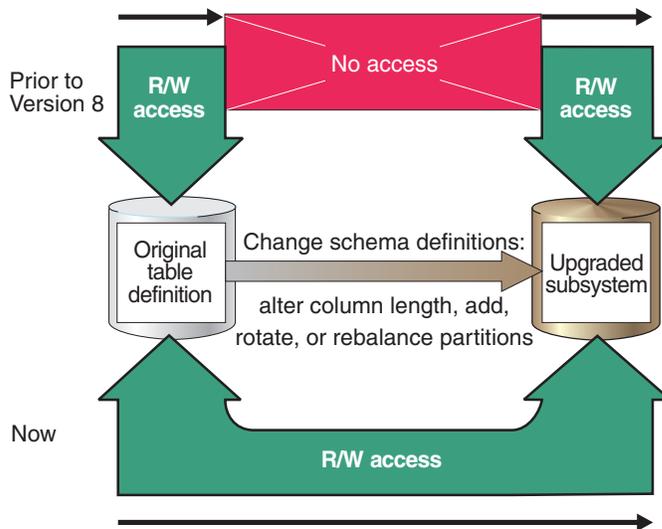


Figure 10. Greater availability with schema evolution

Change column types and lengths

You can change the data type of a table column. The new definition is applied immediately to all data that is in the associated table; when data rows are accessed, those rows are formatted according to the new definition.

Add columns to an index

You can now append columns to the end of an existing index. If you add a column to a table within the same unit of work in which the column is added to the index, the index is available for access immediately.

Add, rotate, or rebalance partitions dynamically

In Version 8, you can add, rotate, or rebalance partitions dynamically while maintaining availability of the partitions:

- You can add partitions at the end of a partitioned table space up to the maximum limit that is defined for the table space. The changes take effect immediately.
- You can reuse (or *rotate*) existing partitions for new data as old data is rolled away. With this support, you can, for example, keep 12 months of data continuously by using only 13 partitions. When you rotate partitions, you delete all the data rows in the oldest (first) partition and then specify a new high boundary for the table space. As a result, the oldest partition rotates to the last logical partition in sequence (the thirteenth partition, in this example) and is ready to hold new data.
- You can use the REBALANCE utility to rebalance partitions without causing a negative impact on availability.

Change the partitioning and clustering of the data in your tables

Version 8 gives you more flexibility for partitioning and clustering the data in your tables:

- You can create a partition without an index. As a result, you can improve performance by eliminating an unnecessary index and the accompanying overhead.
- You can drop a partitioning index or create a table without a partitioning index.

- You can define a clustering order that is different from the partitioning order. For example, if your data is partitioned by month, you can cluster by customer number within each partition.
- You can alter the clustering index.

Greater data availability with data-partitioned secondary indexes

Version 8 of DB2 lets you partition secondary indexes according to the partitioning scheme of the underlying data. *Secondary indexes* are nonpartitioning indexes of partitioned tables. Partitioned secondary indexes are referred to as *data-partitioned secondary indexes*. Data-partitioned secondary indexes can:

- Improve data availability during utility operations that operate at the partition level, such as REORG PART, LOAD PART, and RECOVER PART. For example, the BUILD2 phase and the accompanying outage are eliminated for REORG PART operations.
- Streamline and improve the performance of partition-level operations, such as rotating partitions. For example, the ALTER ROTATE PART operation performs a mass delete without accessing and deleting individual keys.
- Allow secondary indexes to benefit from strategies that reduce overhead costs that are incurred by data sharing.

More flexibility and faster recovery with system-level point-in-time recovery

Enhancements to system-level point-in-time recovery for DB2 provide improved usability, more flexibility, and faster recovery times. As Figure 11 on page 26 shows, you can now recover your data to any point in time, regardless of whether you have uncommitted units of work. As a result, data recovery time improves significantly for large DB2 subsystems that contain more than 30 000 objects, which means that the down time for the subsystem also decreases considerably. Two new utilities provide the vehicle for system-level point-in-time recovery:

- The BACKUP SYSTEM utility provides fast volume-level copies of DB2 databases and logs. It relies on new DFSMSHsm™ services in z/OS Version 1 Release 5 that automatically keep track of the volumes that need to be copied. Using BACKUP SYSTEM is less disruptive than using the SET LOG SUSPEND command for copy procedures because DB2 does not suspend writing to the log. An advantage for data sharing is that BACKUP SYSTEM operates at the group scope, whereas SET LOG SUSPEND operates at the member scope.
- The RESTORE SYSTEM utility recovers a DB2 subsystem to an arbitrary point in time. RESTORE SYSTEM automatically handles any creates, drops, and LOG NO events that might have occurred between the backup and the recovery point in time.

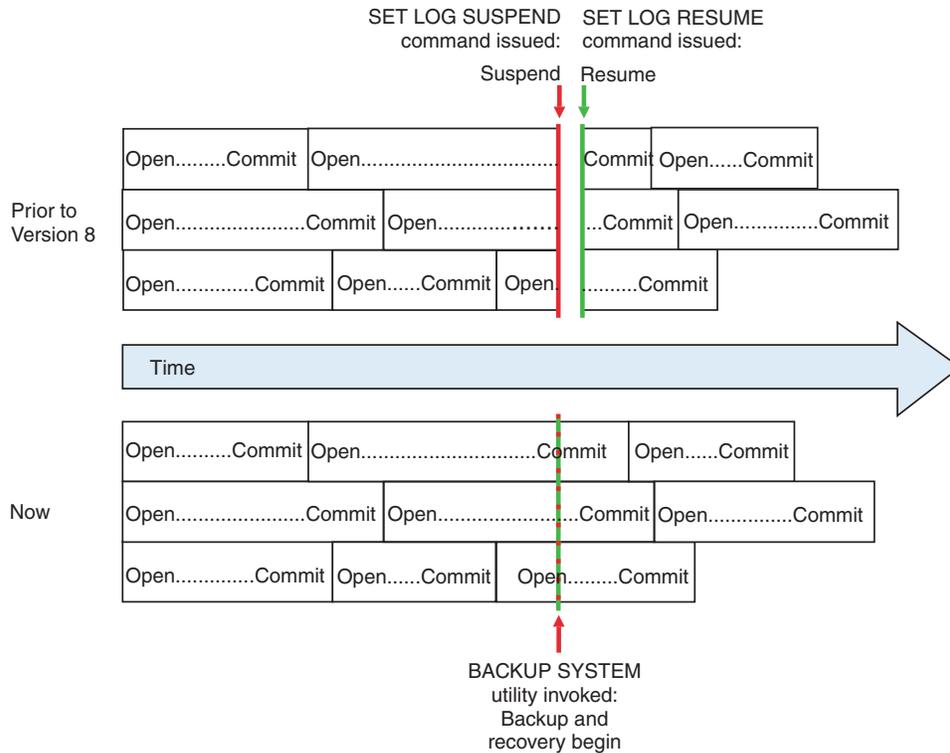


Figure 11. More flexibility and faster recovery with enhancements to system-level point-in-time recovery

Improved utility functions

Many of the Version 8 utility enhancements support other fundamental changes in Version 8 that have been discussed previously:

- Long names (see “Extended limits for names and SQL statements” on page 11)
- Unicode (see “Significant support for Unicode and long names in the DB2 catalog” on page 12)
- 64-bit virtual addressing (“64-bit virtual storage” on page 17)
- Data-partitioned secondary indexes (see “Greater data availability with data-partitioned secondary indexes” on page 25)
- System-level point-in-time recovery (see “More flexibility and faster recovery with system-level point-in-time recovery” on page 25)

Other utility changes include:

- “Online REORG utility”
- “LOAD and UNLOAD utilities” on page 27
- “RUNSTATS distribution statistics” on page 27
- “Autonomic restart” on page 27

Online REORG utility

In addition to REORG changes that support data-partitioned secondary indexes, as discussed in “Greater data availability with data-partitioned secondary indexes” on page 25, you can now:

- Specify the SHRLEVEL CHANGE option for a REORG TABLESPACE DISCARD operation.

- Reorganize DB2 catalog table spaces that have links if you specify SHRLEVEL REFERENCE.
- Specify the SCOPE PENDING keyword to reorganize only the partitions that are in a REORG-pending state (REORP) or an advisory REORG-pending (AREO) state for a specific table space or a partition range.

Online CHECK INDEX utility

Enhancements to the online CHECK INDEX utility in Version 8 increase the availability of your read-only data. New options let you specify:

- The number of seconds that the utility waits when draining a table space or an index
- The maximum number of retries that are to be attempted
- The minimum duration, in seconds, between retries

LOAD and UNLOAD utilities

Prior to Version 8, you could not load a delimited input file into DB2 or unload a delimited output file from DB2. A *delimited file* is a sequential file that contains row and column delimiters.

Now, in Version 8, you can use the LOAD utility to load into DB2 a delimited input file from another relational database. What's more, you do not need to write a program that converts the data into the correct format, or use INSERT processing and give up the performance advantages of the LOAD utility.

In addition, you now can use the UNLOAD utility to unload a delimited output file from DB2 to one or more files that are stored outside of DB2. You can then load the data into another DB2 database in a z/OS environment or on other operating systems, or import the data into an application in another relational database.

RUNSTATS distribution statistics

Skewed data distributions are responsible for a high proportion of performance problems with DB2 queries, especially in ad hoc queries. Symptoms of these problems include join sequences that are not optimal, too much synchronous I/O, and long response times. In addition, the ability of DB2 to make optimal decisions about table join order and table join methods can be weakened if the distribution of your data is asymmetrical, and you do not have distribution statistics on non-leading indexed columns or non-indexed columns.

Improvements to the RUNSTATS utility in Version 8 let you collect distribution statistics for non-leading indexed columns and non-indexed columns so that DB2 can use these statistics to select better access paths.

Autonomic restart

In Version 8, restarting utility jobs is easier because you no longer need to add the RESTART or RESTART(PHASE) parameters to a utility job. DB2 attempts to restart utility jobs that can be restarted online regardless of whether the RESTART keyword is specified.

Enhanced LPL recovery processing

DB2 inserts entries for pages that are logically in error in a logical page list (LPL). Version 8 of DB2 UDB for z/OS offers the following enhancements to LPL recovery to improve usability, serviceability, availability, and performance:

- Automatic recovery of LPL pages: To avoid manual intervention for LPL recovery through the START DATABASE command or the RECOVER utility, DB2, in most cases, automatically initiates an LPL recovery processor to recover pages as they are added to the LPL.
- Less-disruptive LPL recovery: The LPL recovery processor (by way of the START DATABASE command or the new automatic LPL recovery feature), makes a write claim instead of a drain on the object that is being recovered. As a result, good pages in the object are available to SQL users, and performance is improved because the claim is less disruptive than a drain.

Change more parameters online without recycling DB2

Version 7 provided support that let you change a set of subsystem parameters online without recycling DB2. Version 8 expands that set to include many more subsystem parameters that can be changed online. For example, you can now change authorization IDs for the system operators or the system administrator, change the sort pool size, and change the timeout limits for threads.

Increased maximum number of active and archive log data sets

In Version 8, an increase in the maximum number of active and archive log data sets per log copy increases recovery performance and scalability as the volume of your data increases.

The maximum number of archive log volumes that are recorded in the BSDS is increased from 1000 to 10 000 volumes per log copy. With a larger maximum number of archive log volumes per log copy, you can avoid taking frequent image copies.

The maximum number of active log data sets is increased from 31 to 93 per log copy. Recovery performance is improved because reading active logs is much faster than reading archive logs, and the additional active log capacity can decrease the necessity to read archive logs in some cases. Also, increasing the number of active logs can help avoid a full log condition if the offload task stalls or temporarily falls behind.

Autonomic space allocation

Version 8 provides space allocation enhancements to improve performance, increase data availability, and limit the occurrence of outages caused by lack of space. DB2 can now calculate the amount of space to allocate to secondary extents by using a sliding-scale algorithm. The first 127 extents are allocated in increasing size, and the remaining extents are allocated based on the initial size of the data set. This approach has several advantages:

- It minimizes the potential for wasted space by increasing the size of secondary extents gradually.
- It prevents very large allocations for the remaining extents, which would likely cause fragmentation.
- It does not require users to specify secondary space allocation values when creating and altering table spaces and index spaces.
- It allows you, in theory, to always reach the maximum data set size without running out of secondary extents.

Chapter 3. Leveraging your enterprise for e-business

DB2 UDB for z/OS is more than a large storehouse for your enterprise data. Through a combination of powerful base functions and optional features, IBM helps you leverage your enterprise for e-business and maintain your competitive edge. By using these functions and features, you can efficiently and effectively access, analyze, integrate, customize, manage, summarize, and share data and applications throughout your enterprise. Read more about these functions, features, and new offerings in the following sections:

- “Accessing your enterprise data”
- “Boosting application development productivity with DB2 Development Center” on page 33
- “Customizing and extending your enterprise data with XML Extender” on page 34
- “Managing your enterprise” on page 35
- “DB2 and IMS Tools for your database server” on page 38

Accessing your enterprise data

DB2 UDB for z/OS delivers query support through the DB2 QMF™ family of products and works well with other IBM query tools such as Web Query and Office Connect.

DB2 QMF family of products

DB2 Query Management Facility (DB2 QMF) is a family of integrated tools that offers a total solution to your enterprise query and reporting needs. These tools let you work with DB2 data from any DB2 UDB family member. With these tools, you can access large amounts of data, share central repositories of queries and reports, and publish reports to the Web for viewing with a browser.

The DB2 QMF family of products includes:

- “DB2 QMF for WebSphere” on page 30 (a new offering)
- “DB2 QMF for Windows” on page 31
- “DB2 QMF for TSO/CICS” on page 31
- “DB2 QMF High Performance Option” on page 32

Highlights of Version 8 of DB2 QMF

The following list describes just a few of the many enhancements that are provided in Version 8 of DB2 QMF:

- DB2 QMF for WebSphere, a tool that turns any Web browser into a zero-maintenance, thin client for visual on demand access to enterprise DB2 UDB data.
- Support for DB2 Cube Views in DB2 QMF for WebSphere and DB2 QMF for Windows. DB2 Cube Views is the latest generation of online analytic processing (OLAP) support that lets you store and manipulate OLAP metadata in the DB2 catalog.
- Full support for functional enhancements to Version 8 of DB2 UDB for z/OS, including long names, Unicode, and enhancements to SQL.
- New packaging options that offer simplified selection and pricing to more easily match the information needs of any enterprise, no matter what platforms are used by the end user or by DB2 UDB.

For a complete list of enhancements to the DB2 QMF family, see the DB2 QMF, Version 8 software announcement at www.ibm.com/qmf/

DB2 QMF for WebSphere

DB2 QMF for WebSphere, a new offering in Version 8, is a browser-based portal to enterprise DB2 UDB data. DB2 QMF for WebSphere supports every IBM WebSphere platform and requires no additional plug-in support or applications to turn ordinary Web browsers into zero-maintenance thin, clients for on demand access to enterprise DB2 UDB data. Through intranet or Internet connections, users can share many of the query, reporting, and OLAP functions that are found in DB2 QMF for Windows and DB2 QMF for TSO/CICS. (See Figure 12.)

DB2 QMF for WebSphere supports Version 8 Release 1 of IBM DB2 Cube Views, which provides easy access to data and lets you perform real-time data analysis. What's more, no extensive knowledge of OLAP is required. With support for DB2 Cube Views, you can drag multiple objects onto predefined layouts to create dynamic OLAP reports in minutes, from multiple levels of summary reports to detailed analytic reports. In addition, you can easily format and print long and complex reports on your local printer. Speed and accuracy of data analysis is critical in a highly competitive on demand environment—DB2 QMF and DB2 Cube Views deliver both. For more information about DB2 Cube Views, see www.ibm.com/db2/db2md/

DEPT	JOB	ID	NAME	YEARS	SALARY	COMM
-10	+MGR			12	\$20,865.86	
	All values for 10			12	\$20,865.86	
-15	-CLERK	110	NGAN	5	\$12,508	\$207
		170	KERMISCH	4	\$12,259	\$110
				5	\$12,383.35	\$206.60
	+MGR			10	\$20,659.80	
+20	+SALES			7	\$16,502.83	\$1,152.00
	All values for 15			10	\$15,482.33	\$1,152.00
+38	All values for 20			8	\$16,071.53	\$612.45
+42	All values for 38			6	\$15,457.11	\$846.55
	All values for 42			7	\$14,592.26	\$1,386.70

Figure 12. A Database Explorer results table in DB2 QMF for WebSphere that shows the QMF objects that reside on the database server

For a complete list of enhancements to DB2 QMF for WebSphere, see the DB2 QMF, Version 8 software announcement at www.ibm.com/qmf/

DB2 QMF for Windows

DB2 QMF for Windows now provides significant new data visualization capabilities and many other enhancements including:

- Support for Version 8 Release 1 of IBM DB2 Cube Views, which provides new visual query and report building capabilities as discussed in “DB2 QMF for WebSphere” on page 30.
- DB2 QMF Visionary, a feature for building powerful visual information applications such as executive dashboards. With DB2 QMF Visionary Studio, a subcomponent of DB2 QMF Visionary, you can rapidly build and deploy (without programming or compiling) highly intuitive applications that let users visually zoom into, analyze, and summarize data. For example, Figure 13 shows a dual graph report that can be used for an executive briefing.
- Database Explorer for easily browsing and identifying database assets and other objects that they might reference.
- New visual query and report-building capabilities.

For a complete list of Version 8 enhancements to DB2 QMF for Windows, see the DB2 QMF, Version 8 software announcement at www.ibm.com/qmf/

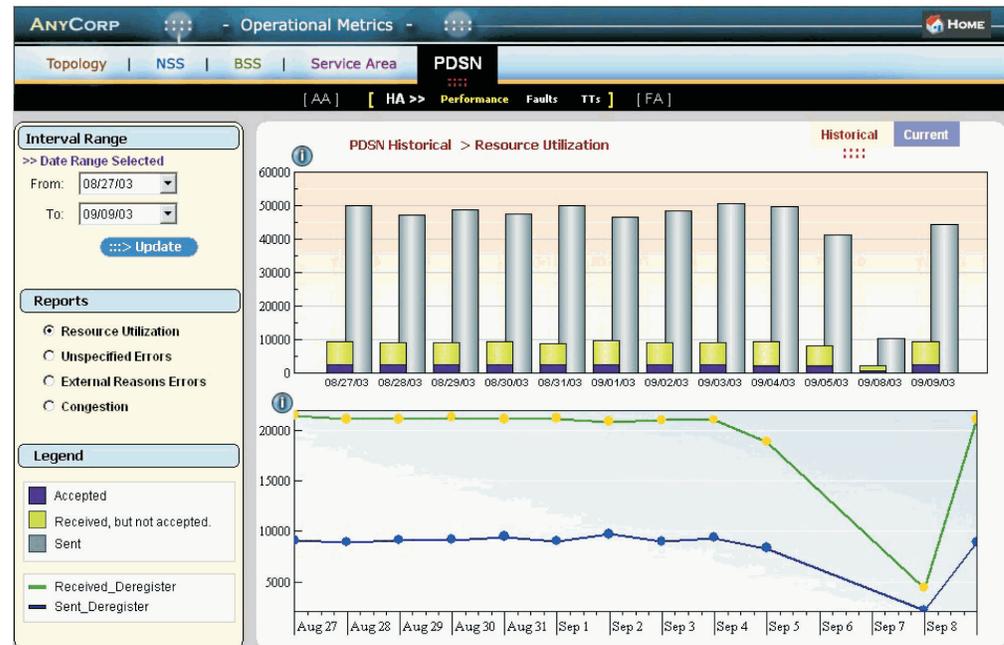


Figure 13. A dual graph report created in DB2 QMF Visionary

DB2 QMF for TSO/CICS

DB2 QMF for TSO/CICS is the leading z/OS-based query environment that provides access across all DB2 UDB family clients and is ready to support your needs from day one. Version 8 provides significant enhancements to core functions, including:

- Support for enhancements to Version 8 of DB2 UDB for z/OS, including long names
- The ability to execute stored procedures and generate queries into stored procedures
- Fully integrated support for CLOB, BLOB, and DBCLOB data types

For a complete list of Version 8 enhancements to DB2 QMF for TSO/CICS, see the DB2 QMF, Version 8 software announcement at www.ibm.com/qmf/

DB2 QMF High Performance Option

DB2 QMF High Performance Option (HPO) adds a suite of administrative and application building capabilities to DB2 QMF for TSO/CICS. Enhancements to Version 8 include:

- Automatic generation of stored procedures from DB2 QMF queries
- Support for long names in the DB2 QMF for TSO/CICS object catalog

For a complete list of DB2 QMF HPO enhancements, see the DB2 QMF, Version 8 software announcement at www.ibm.com/qmf/

New packaging options for DB2 QMF

DB2 QMF offers three packaging options: DB2 QMF Enterprise Edition, DB2 QMF Distributed Edition, and DB2 QMF Classic Edition. Both DB2 QMF Distributed Edition and DB2 QMF Classic Edition offer subsets of DB2 QMF Enterprise Edition.

DB2 QMF Enterprise Edition: DB2 QMF Enterprise Edition provides the entire DB2 QMF family of technologies that enables enterprise-wide business information across end user platforms and database platforms. DB2 QMF Enterprise Edition consists of the following components:

- DB2 QMF for TSO/CICS
- DB2 QMF High Performance Option
- DB2 QMF for Windows
- DB2 QMF for WebSphere

DB2 QMF Distributed Edition: DB2 QMF Distributed Edition provides components to support end users that access enterprise databases solely through Web or Windows clients. DB2 QMF Distributed Edition consists of the following components:

- DB2 QMF for Windows
- DB2 QMF for WebSphere
- DB2 QMF Visionary

DB2 QMF Classic Edition: DB2 QMF Classic Edition supports end users that access DB2 UDB for z/OS databases solely from traditional mainframe terminals and emulators, including IBM Host On Demand. DB2 QMF Classic Edition consists of DB2 QMF for TSO/CICS.

Integrating your zSeries and iSeries data with DB2 Connect

DB2 Connect™ is an industry-leading solution that integrates zSeries and iSeries™ data with client-server and Web applications. Version 8 Release 1 of DB2 Connect offers the following enhancements in manageability, availability, performance, scalability, and application development:

- Enhancements to manageability include:
 - New tools, Health Monitor and Health Center, that alert you to potential system-health issues before they become real problems that affect performance
 - Enhanced DB2 Control Center functions for DB2 and z/OS administrators
 - Simplified cross-system data movement and IMS system administration
- Improvements to application availability through additional enhancements to the Connection Concentrator function

- Improvements to application performance and scalability include:
 - Improved resource utilization, higher scalability, and better performance
 - A 64-bit version that supports modern server hardware and operating environments
 - Enhancements to drivers for OLE DB, ODBC, JDBC, and SQLJ
- Improvements to application development include:
 - Improved drivers for popular application programming interfaces (APIs)
 - Enhanced function and ease of building DB2 server-side objects (stored procedures, functions, and triggers) and improved integration with popular development environments
 - Support for Web services

Boosting application development productivity with DB2 Development Center

The new DB2 Development Center, an element in the DB2 Management Clients Package, extends the capabilities of the DB2 Stored Procedure Builder that was available before Version 8. The DB2 Development Center also boosts the efficiency of your application development and deployment cycle. DB2 Development Center continues to provide an easy-to-use development environment for creating, installing, and testing stored procedures. (See Figure 14 on page 34.) The enhanced functionality of DB2 Development Center supports:

- Rapid iterative development of SQL and Java stored procedures through easy-to-use wizards that provide code-generation templates that you can customize and that let you insert automatic code fragments.
- Efficient project management through new and advanced import, export, and deployment features and wizards.
- Read-only access to stored procedures, user-defined functions, triggers, tables, and views through a new server view. The server view lets you look at your live server objects separately from your project objects that are shown in the project view.
- Online (connected) and offline (disconnected) database connection modes with connection pooling for better resource management.
- Advanced build and authorization options for z/OS, including primary SQL ID and additional SQL IDs for secondary authorization identification, package owner, and build owner.
- Support for test runs of stored procedures in any language or function such as COBOL and PL/I.
- Support for persistent test settings, including parameters, options, and pre-execution and post-execution SQL scripts. Because they are persistent, test settings can be used again for subsequent testing.
- An integrated debugging tool, SQL Debugger, that lets you remotely debug SQL stored procedures that execute on DB2 servers. With SQL Debugger, you can observe the execution of SQL procedure code, set break points for lines, and view or modify variable values.

DB2 Development Center also supports a fully integrated set of DB2 Development Add-Ins for Microsoft® Visual Studio 6.0 (for Visual Basic, Visual C++, and Visual InterDev). For example:

- Microsoft Visual SourceSafe enables rapid iterative development of server-side stored procedures and client-side generation and integration of Microsoft ActiveX Data Objects (ADO) code.
- DB2 Development Add-Ins for Microsoft Visual Studio .NET is a set of tightly integrated application development and administration tools that are designed for DB2 Universal Database. The Add-Ins integrate into the Visual Studio .NET development environment so that .NET developers can work within their integrated development environment to easily access DB2 data.

Key benefits include:

- Support for client applications (both desktop and Web-based applications) to use .NET to access remote DB2 servers
- A tool for building stored procedures that lets any application programmer easily develop and test stored procedures with DB2 UDB for z/OS without prior zSeries skills or knowledge.

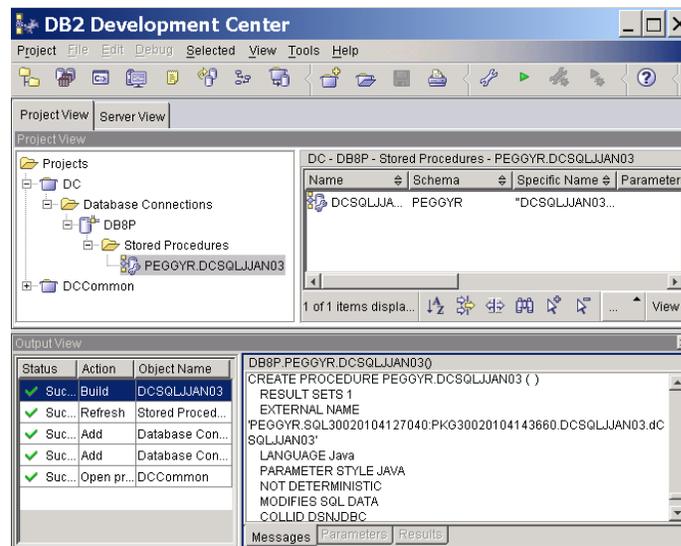


Figure 14. A project view as displayed by DB2 Development Center

Customizing and extending your enterprise data with XML Extender

Staying competitive in the e-business world requires powerful tools to facilitate the development and interchange of data. With the emergence of XML as the accepted standard for data interchange, DB2 UDB for z/OS keeps pace with industry standards by supporting XML functions in DB2 and by managing and taking advantage of XML data through the XML Extender product, which is shipped with DB2 UDB for z/OS.

Version 8 of DB2 UDB for z/OS XML Extender offers new user-defined functions that:

- Validate XML documents by using XML schemas
- Transform XML documents to other formats and results by using XSLT style sheets

XML Extender also offers:

- Increased limit sizes for many objects (for example, the maximum number of characters in a column name that is specified in a default view)
- Enhancements to XML collections that include:

- Two new stored procedures that eliminate the need for temporary or permanent tables for composed documents, which simplifies programming in a multi-user environment, reduces the instruction path length, and improves throughput
- The ability to store, in temporary tables, the XML documents that are returned from a composition stored procedure
- Improved usability of document access definition (DAD) file syntax

Managing your enterprise

Version 8 of DB2 UDB for z/OS offers several tools that help you manage your data sets, perform administration tasks for your remote servers, configure your subsystem in a z/OS environment, and perform capacity planning tasks:

- “Managing DB2 UDB for z/OS data sets from the DB2 Control Center”
- “DB2 Administration Server” on page 36
- “Using msys for Setup DB2 Customization Center to customize DB2” on page 36
- “DB2 Visual Explain” on page 37
- “DB2 Estimator” on page 38

Managing DB2 UDB for z/OS data sets from the DB2 Control Center

The IBM DB2 Control Center provides a common interface for managing DB2 family databases on different operating systems. (See Figure 15 on page 36.) From the DB2 Control Center, you can run DB2 commands, create DDL statements, and execute DB2 utilities. In Version 8, DB2 Control Center support for DB2 UDB for z/OS has been enhanced significantly to support operation in a z/OS environment. For example, the DB2 Control Center now provides support for:

- A subsystem cloning wizard that generates the job control language (JCL) that is required to homogeneously copy (clone) an entire subsystem to a target subsystem
- A stored procedure that schedules parallel execution of an online utility and optimizes the elapsed time and system load of utility execution on a large number of objects
- Integrated DB2 batch processing functions: build JCL, create JCL, and custom JCL

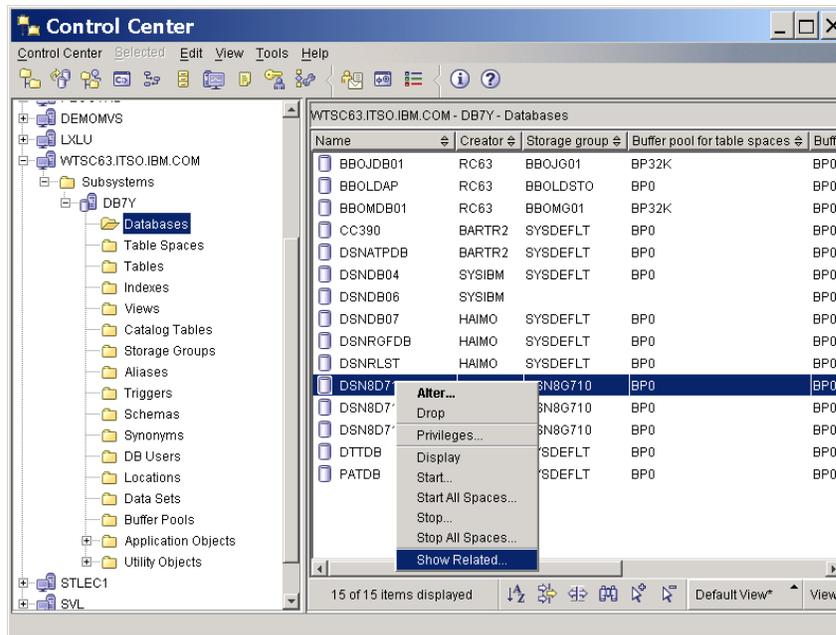


Figure 15. DB2 Control Center provides a common interface for managing your DB2 family databases.

DB2 Administration Server

The DB2 Administration Server is now available for the zSeries. The DB2 Administration Server is required by and supports the following administrative tasks for DB2 Control Center, DB2 Development Center, and DB2 and IMS Replication Center:

- Remote administration of DB2 servers
- Job management, including setting the schedule for running DB2 and user-defined operating-system command scripts

Using msys for Setup DB2 Customization Center to customize DB2

The IBM z/OS Managed System Infrastructure for Setup DB2 Customization Center (msys for Setup) is part of a major z/OS ease-of-use initiative. msys for Setup simplifies system configuration in the z/OS environment through workstation-based configuration windows. The configuration windows provide default values and recommendations for best practices wherever possible. To set up z/OS and its products, you specify a configuration through the windows, and msys for Setup applies the configuration to the system automatically. (See Figure 16 on page 37.) Version 8 of DB2 UDB for z/OS joins the growing number of products that you can customize with msys for Setup. msys for Setup DB2 Customization Center simplifies the task of customizing a DB2 subsystem on a z/OS operating system. A set of wizards provides guidance for specifying settings for DB2 subsystem parameters (DSNZPARMs) and reduces the amount of data that you need to enter to set up and configure DB2. After you complete the initial setup and configuration of DB2, you can view and update the parameters and other information through notebooks.

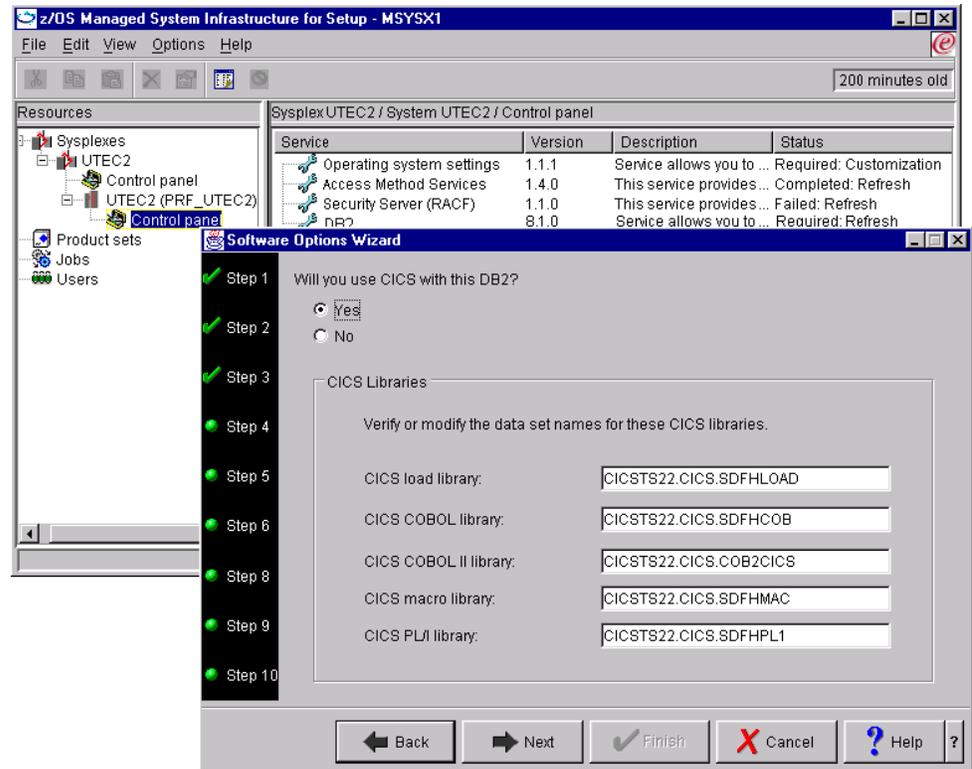


Figure 16. An example of a Software Options Wizard in an msys for Setup DB2 Customization Center session

DB2 Visual Explain

DB2 Visual Explain eliminates the need to manually interpret plan table output by providing graphical depictions of the access plans that DB2 chooses for your SQL queries and statements. (See Figure 17 on page 38.) DB2 Visual Explain provides robust function that can generate a complete access plan graph that includes each SQL statement and complex queries that consist of multiple subqueries. New functions in Version 8 of DB2 Visual Explain provide:

- More context-sensitive tuning suggestions. You can link to suggestions directly from the graph, or you can link from the graph to additional statistics and descriptions for each object or operation that is used in the access plan.
- The capability to display single or multiple query blocks on one graph.
- The capability to catalog and uncatalog databases on your local machine.
- The capability to run a query and view the formatted results.

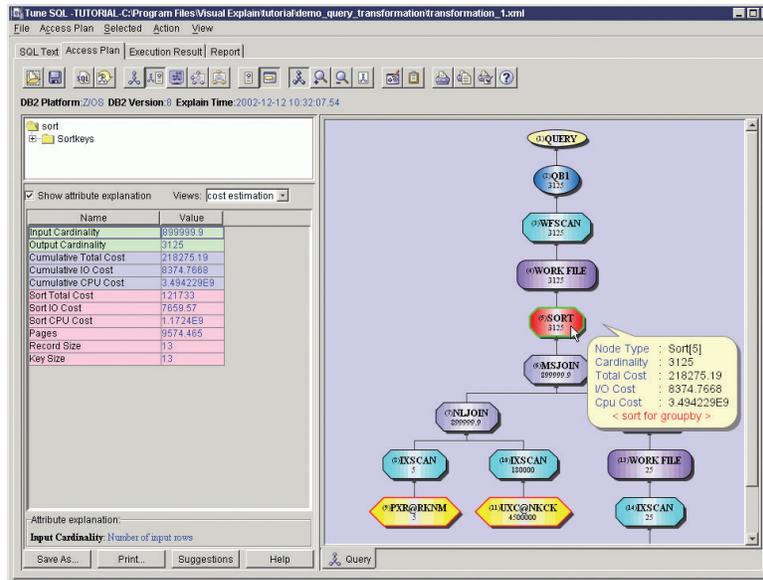


Figure 17. An access plan view with detailed cost estimations as displayed by DB2 Visual Explain

DB2 Estimator

DB2 Estimator is an easy-to-use, stand-alone tool for estimating the storage needs and performance of applications for DB2 UDB for z/OS. You can run DB2 Estimator on your desktop personal computer or take it with you on your laptop computer. DB2 Estimator is delivered on the Web. Watch for new releases at www.ibm.com/software/db2zos/estimate/

DB2 and IMS Tools for your database server

IBM has grouped its DB2 and IMS tools into six functional areas to match your database and system requirements:

- Administration tools to maximize database availability
- Utilities to handle a wide variety of database tasks automatically
- Performance management tools to keep systems and databases running smoothly
- Recovery tools to assist with unique backup and recovery situations
- Replication tools to improve the availability of copied data
- Application management tools to help you create and operate application subsystems

Together, the array of tools that IBM offers can help you meet performance standards and maintain IT costs. You can combine a selection of tools from any category to build a complete, flexible and affordable solution that is tailored to meet your specific needs. A wide variety of tools that work with DB2 UDB for z/OS are available and ready to support all major new Version 8 functions.

#

For the most current information about tools that support Version 8, see ibm.com/software/data/db2imstools

Important

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

Glossary

The following terms and abbreviations are defined as they are used in the DB2 library.

A

automatic query rewrite. A process that examines an SQL statement that refers to one or more base tables, and, if appropriate, rewrites the query so that it performs better. This process can also determine whether to rewrite a query so that it refers to one or more materialized query tables that are derived from the source tables.

C

clustering index. An index that determines how rows are physically ordered (*clustered*) in a table space. If a clustering index on a partitioned table is not a partitioning index, the rows are ordered in cluster sequence within each data partition instead of spanning partitions. Prior to Version 8 of DB2 UDB for z/OS, the partitioning index was required to be the clustering index.

D

DAD. See *Document access definition*.

data-partitioned secondary index (DPSI). A secondary index that is partitioned. The index is partitioned according to the underlying data.

document access definition (DAD). Used to define the indexing scheme for an XML column or the mapping scheme of an XML collection. It can be used to enable an XML Extender column of an XML collection, which is XML formatted.

DPSI. Data-partitioned secondary index.

dynamic cursor. A named control structure that an application program uses to change the size of the result table and the order of its rows after the cursor is opened. Contrast with *static cursor*.

F

forest. An ordered set of subtrees of XML nodes.

H

host variable array. An array of elements, each of which corresponds to a value for a column. The

dimension of the array determines the maximum number of rows for which the array can be used.

I

index-controlled partitioning. A type of partitioning in which partition boundaries for a partitioned table are controlled by values that are specified on the CREATE INDEX statement. Partition limits are saved in the LIMITKEY column of the SYSIBM.SYSINDEXPART catalog table.

L

location alias. Another name by which a database server identifies itself in the network. Applications can use this name to access a DB2 database server.

M

materialized query table. A table that is used to contain information that is derived and can be summarized from one or more source tables.

N

nonpartitioned index. An index that is not physically partitioned. Both partitioning indexes and secondary indexes can be nonpartitioned.

nonpartitioned secondary index (NPSI). An index on a partitioned table space that is not the partitioning index and is not partitioned.

nonpartitioning index. See *secondary index*.

NPSI. See *nonpartitioned secondary index*.

P

partitioned index. An index that is physically partitioned. Both partitioning indexes and secondary indexes can be partitioned.

partitioning index. An index in which the leftmost columns are the partitioning columns of the table. The index can be partitioned or nonpartitioned.

R

rowset. A set of rows for which a cursor position is established.

rowset-positioned access. The ability to retrieve multiple rows from a single FETCH statement.

row-positioned access. The ability to retrieve a single row from a single FETCH statement.

S

schema. (1) The organization or structure of a database. (2) A logical grouping for user-defined functions, distinct types, triggers, and stored procedures. When an object of one of these types is created, it is assigned to one schema, which is determined by the name of the object. For example, the following statement creates a distinct type T in schema C:

```
CREATE DISTINCT TYPE C.T ...
```

secondary index. A nonpartitioning index on a partitioned table.

sensitive cursor. A cursor that is sensitive to changes that are made to the database after the result table has been materialized.

sequence. A user-defined object that generates a sequence of numeric values according to user specifications.

source table. A table that can be a base table, a view, a table expression, or a user-defined table function.

static cursor. A named control structure that does not change the size of the result table or the order of its rows after an application opens the cursor. Contrast with *dynamic cursor*.

T

table-controlled partitioning. A type of partitioning in which partition boundaries for a partitioned table are controlled by values that are defined in the CREATE TABLE statement. Partition limits are saved in the LIMITKEY_INTERNAL column of the SYSIBM.SYSTABLEPART catalog table.

transient XML data type. A data type for XML values that exists only during query processing.

tree structure. A data structure that represents entities in nodes, with a most one parent node for each node, and with only one root node.

X

XML. See *Extensible Markup Language*.

XML attribute. A name-value pair within a tagged XML element that modifies certain features of the element.

```
# XML element. A logical structure in an XML
# document that is delimited by a start and an end tag.
# Anything between the start tag and the end tag is the
# content of the element.
```

XML node. The smallest unit of valid, complete structure in a document. For example, a node can represent an element, an attribute, or a text string.

XML publishing functions. Functions that return XML values from SQL values.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

CICS	iSeries
DataPropagator	OS/390
DB2	Parallel Sysplex
DB2 Connect	QMF
DB2 Universal Database	S/390
DFSMSHsm	WebSphere
Distributed Relational Database Architecture	z/Architecture
DRDA	z/OS
IBM	zSeries
IMS	

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Index

Numerics

- 64-bit virtual storage
 - description 17
 - illustration 17

A

- access paths, graphical depictions of 37
- accounting aggregation, improved control for 22
- accounting, package-level, improved 23
- active log data sets, increased maximum 28
- ActiveX Data Objects (ADO) code, client-side generation of 33
- aggregate functions 11
- archive log data sets, increased maximum 28
- ASCII, Unicode, and EBCDIC columns in the same SQL statement 13
- attributes of identity columns, altering 6
- autonomic restart, utility support for 27
- autonomic space allocation 28
- availability enhancements 23

B

- BACKUP SYSTEM utility 25
- backward index scans 19
- built-in functions, data encryption and decryption 8

C

- catalog, DB2
 - long names 13
 - Unicode, support for 13
- CHECK INDEX utility enhancements 27
- CLIENT_USERID special register 10
- CLIENT_ACCTNG special register 9
- CLIENT_APPLNAME special register 9
- CLIENT_WRKSTNNAME special register 10
- clustering index 24
- column lengths, changing 24
- column names, longer 11
- column types, changing 24
- columns, adding to an index 24
- common client, components
 - administrative client 13
 - C common client for ODBC 13
 - Java common client for SQLJ and JDBC 13
- common table expression, recursive 7
- common table expressions 7
- connectivity improvements 15
- coupling facility, reduced lock propagation 22
- CURRENT PACKAGE PATH special register 7
- cursors, dynamic scrollable 6
- customizing DB2 with msys for Setup DB2 Customization Center 36

D

- data encryption built-in function 8
- data encryption tool 9
- data sets, increased maximum of active and archive log 28
- data sharing, reduced overhead costs 22
- data-partitioned secondary indexes 25
- database connection protocols, standardizing 13
- DB2 Administration Server 36
- DB2 and IMS Tools
 - IBM Data Encryption for IMS and DB2 Databases 9
- DB2 catalog, descriptions in 12
- DB2 Connection Concentrator function 32
- DB2 Control Center 35
- DB2 Development Add-Ins for Microsoft Visual Studio .NET 33
- DB2 Development Center
 - DB2 Development Add-Ins for Microsoft Visual Studio .NET 33
 - Microsoft Visual SourceSafe 33
 - SQL Debugger feature 33
- DB2 Estimator 38
- DB2 family compatibility
 - additional input format for timestamp strings 12
 - aggregate functions 11
 - column names, longer 11
 - declared global temporary tables, implicit drop at commit 12
 - descriptions of plans and packages in the DB2 catalog 12
 - extensions to SQL procedure statements 11
 - long names 13
 - network computing, enhancements 13
 - number of tables joined in a FROM clause 11
 - ORDER BY clause for the SELECT INTO statement 12
 - qualified COLUMN names in the INSERT statement 12
 - ROWID columns generated implicitly for LOB columns 12
 - SQL statements, longer 11
 - Unicode 13
- DB2 Stored Procedure Builder
 - See DB2 Development Center
- DB2 Visual Explain 37
- declared global temporary tables, implicit drop at commit 12
- delimited file, definition of 27
- diagnostic information, returning from an SQL statement 7
- DISTINCT keyword, multiple allowed in a query 12
- DISTINCT predicate, compare null values with 8
- distribution statistics for improved optimization 19
- dynamic changes to partitions 24
- dynamic scrollable cursors 6
- dynamic SQL, improved optimization for 19

E

- e-business
 - accessing enterprise data
 - DB2 QMF for TSO/CICS 31
 - DB2 QMF for WebSphere 30
 - DB2 QMF for Windows 30
 - DB2 QMF High Performance Option 32
 - QMF family of products 29

- e-business (*continued*)
 - application development productivity 33
 - customizing and extending enterprise data
 - XML Extender 34
 - XML publishing functions 6
 - DB2 Development Center 33
 - managing your enterprise
 - DB2 Administration Server 36
 - DB2 Control Center 35
 - DB2 Estimator 38
 - IBM DB2 and IMS Tools 38
 - Intelligent Visual Explain 37
 - msys for Setup DB2 Customization Center, use of 36
- EBCDIC, ASCII, and Unicode columns in the same SQL statement 13
- encoding schemes
 - mixed 13
 - ODBC support 16
- encrypted security in distributed computing environments 10
- encryption
 - built-in functions for data encryption and decryption 8
 - data encryption tool 9
 - encrypted security in distributed computing environments 10
- Estimator, DB2 38

F

- fetches, multiple 20
- fullselects in SQL statements, scalar 6

G

- GET DIAGNOSTICS statement 7
- GETVARIABLE built-in function 10
- glossary 41
- GROUP BY clause, expressions in 12

H

- Health Center 32
- Health Center tool 32
- Health Monitor tool 32

I

- IBM Data Encryption for IMS and DB2 Databases 9
- identity columns, altering attributes 6
- index
 - adding columns to 24
 - scans, backward 19
 - secondary, data-partitioned
 - definition 25
 - greater availability with 25
 - variable-length keys, stored 18
- index enhancements
 - backward index scans for avoiding sorts 19
 - capability to index predicates that have mismatched data types 18
 - stored variable-length index keys 18
- index keys, longer 11
- index predicates that have mismatched data types, capability to 18
- INSERT statement, qualifying column names in 12
- inserts, multiple 20

- Intelligent Visual Explain 37
- iSeries data, integrating with zSeries data via DB2 Connect 32

J

- Java
 - integration with DB2 and WebSphere 2
 - Java common client for SQLJ and JDBC 13
- joined tables, number in a single FROM clause 11

L

- limits, extended
 - column names in the INSERT statement, capability to qualify 12
 - column names, longer 11
 - CREATE PROCEDURE statement 11
 - index keys, longer 11
 - joined tables in a FROM clause, more 11
 - partitioned table, size of 20
 - partitions in a partitioned table space, number of 20
 - predicates, longer 11
 - SQL statements, longer and more complex 11
- LOAD utility enhancements 27
- LOB columns, implicitly generated ROWID columns for 12
- lock propagation in coupling facility, reduced 22
- log data sets, increased maximum of active and archive 28
- logical page list recovery 27
- LPL recovery processing 27

M

- manageability tools, Health Monitor 32
- Managed System Infrastructure for Setup DB2 Customization Center (msys for Setup) 36
- materialized query tables 18
- msys for Setup DB2 Customization Center, use of 36
- multilevel security with row-level granularity 9
- multiple fetches and inserts 20

N

- names, long 13
- network computing, enhancements
 - applications, portability of 13
 - connectivity improvements 15
 - remote queries, performance improvement 14
- notices, legal 43

O

- ODBC enhancements 16
- optimization improvements
 - additional distribution statistics 19
 - dynamic SQL 19

P

- package-level accounting, improved 23
- parallel processing of sort operations 20
- parallel processing, cost-based 20
- parameters, changing online 28
- partitioned table, increased size of 20

- partitioning of data, changing 24
- partitions
 - adding 24
 - rebalancing 24
 - rotating 24
- partitions in a partitioned table space, increased number of 20
- password, common, for encryption 8
- performance improvements
 - DRDA applications from z/OS requester 7
 - Java programs that use SQLJ 7
 - remote queries, for 14
 - star-join queries 20
 - stored procedures 7
 - trigger performance 19
 - user-defined functions 7
- portability across DB2 family 13
- predicates
 - longer 11
 - mismatched data types, indexing 18
- predicates, longer 11
- publishing functions, XML 6

R

- rebalancing partitions dynamically 24
- recovery enhancements
 - LPL recovery processing 27
 - system-level point-in-time 25
- recursive common table expression 7
- recursive SQL 7
- recycling DB2 avoiding by changing parameters online 28
- reliability enhancements 23
- remote queries, performance improvements 14
- REORG utility enhancements 26
- resource utilization
 - stored procedures 22
 - user-defined functions 22
- restart, utility support for autonomic 27
- RESTORE SYSTEM utility 25
- rotating partitions dynamically 24
- row-level granularity in multilevel security 9
- RUNSTATS utility
 - additional distribution statistics for optimization 19
- RUNSTATS utility enhancements 27

S

- scalar fullselects in SQL statements 6
- schema evolution
 - adding columns to an index 24
 - adding, rotating, or rebalancing partitions 24
 - changing column types and lengths 24
 - choosing a clustering index 24
 - overview 23
- scrollable cursors, dynamic 6
- secondary index, data partitioned 25
- secondary space allocation, autonomic 28
- security
 - built-in functions for data encryption and decryption 8
 - CLIENT_ACCTNG special register 9
 - CLIENT_APPLNAME special register 9
 - CLIENT_USERID special register 10
 - CLIENT_WRKSTNNAME special register 10
 - data encryption tool 9

- security (*continued*)
 - encrypted security in distributed computing environments 10
 - GETVARIABLE built-in function 10
 - multilevel, with row-level granularity 9
 - session variables 10
 - system users, easier identification of 9
- SELECT FROM INSERT statement 5
- SELECT INTO statement, ORDER BY clause in 12
- sequence data object, definition 6
- sequential numbers, unique 6
- serviceability enhancements 23
- session variables, security policy enforcement 10
- sort operations
 - avoiding with backward index scans 19
 - parallelism, increased 20
- space allocation, autonomic 28
- special registers
 - CLIENT_ACCTNG 9
 - CLIENT_APPLNAME 9
 - CLIENT_USERID 10
 - CLIENT_WRKSTNNAME 10
 - CURRENT PACKAGE PATH 7
- SQL
 - backward index scans 19
 - column names, longer 11
 - common table expressions 7
 - CURRENT PACKAGE PATH special register 7
 - DISTINCT keywords in queries, multiple 12
 - DISTINCT predicate, compare null values with 8
 - dynamic scrollable cursors 6
 - dynamic SQL, improved optimization for 19
 - enhancements 5
 - extensions to SQL procedure statements 11
 - GET DIAGNOSTICS statement 7
 - GROUP BY clause, support for expressions in 12
 - index keys, longer 11
 - limit for joining tables in a FROM clause 11
 - materialized query tables 18
 - multiple fetches and inserts 20
 - ORDER BY clause for the SELECT INTO statement 12
 - qualified column names 12
 - recursive 7
 - scalar fullselects in SQL statements 6
 - SELECT FROM INSERT statement 5
 - statements, longer and more complex 11
 - unique sequential numbers 6
 - XML publishing functions 6
- SQL Debugger 33
- star-join queries, performance enhancements 20
- statistics, non-leading indexed columns and non-indexed columns 27
- storage, estimating 38
- stored procedures enhancements 22
- stored procedures, developing 33
- stored procedures, server-side development of 33
- subsystem parameters, changing 28
- system-level point-in-time recovery 25

T

- tables, changing clustering of data in 24
- tables, changing partitioning of data in 24
- tables, number of joined in a single FROM clause 11
- timestamp strings, additional input format for 12
- trigger performance, improvements to 19

U

- Unicode
 - support for, enhanced 13
- Unicode, ASCII, and EBCDIC columns in the same SQL statement 13
- Universal Database Driver for z/OS 15
- UNLOAD utility enhancements 27
- user-defined function enhancements 22
- utility enhancements
 - autonomic restart, support for 27
 - BACKUP SYSTEM utility 25
 - CHECK INDEX utility 27
 - LOAD utility 27
 - REORG utility 26
 - RESTORE SYSTEM utility 25
 - RUNSTATS 27
 - UNLOAD 27

V

- variable-length index keys 18
- virtual storage, 64-bit 17
- Visual Explain 37

W

- WebSphere, integration with DB2 and Java 2

X

- XML Extender 34
- XML publishing functions 6

Z

- z/OS Application Connectivity to DB2 for z/OS 15
- z/OS msys for Setup DB2 Customization Center 36
- zSeries architecture
 - 64-bit virtual storage 17
 - DB2 integration with 1
 - integrating iSeries data with DB2 Connect 32



Program Number: 5625-DB2

Printed in USA

GC18-7428-01

