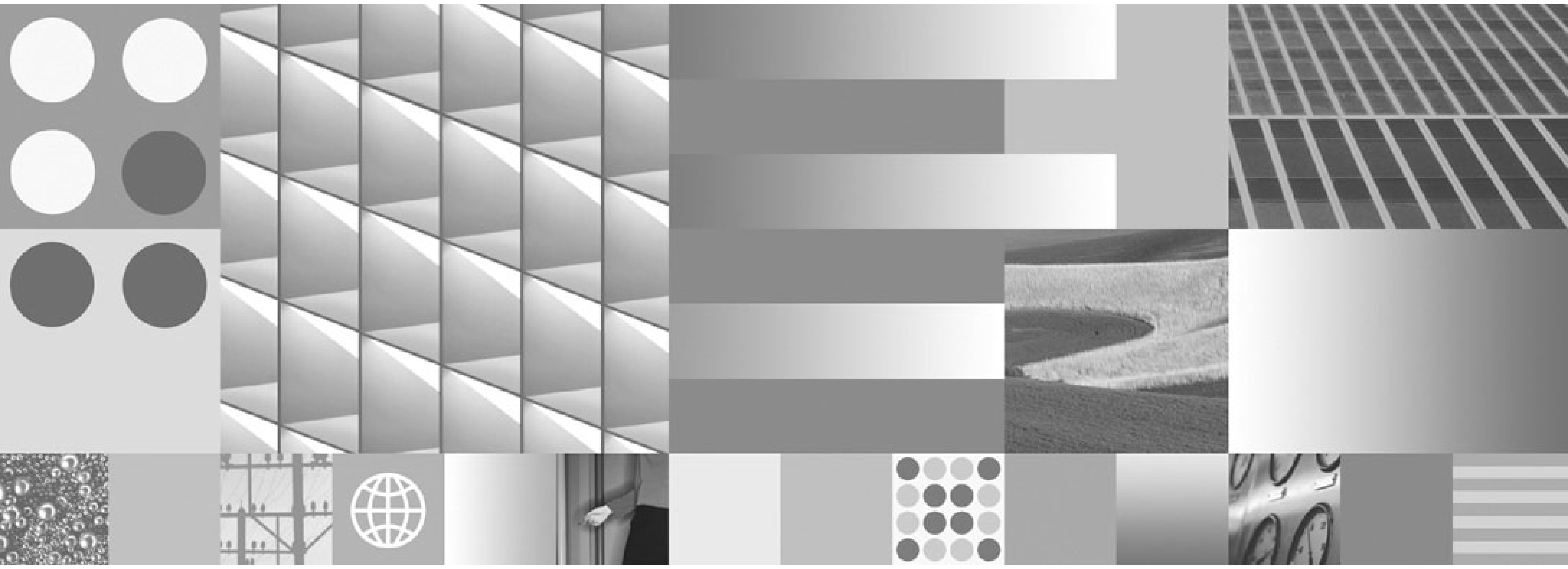
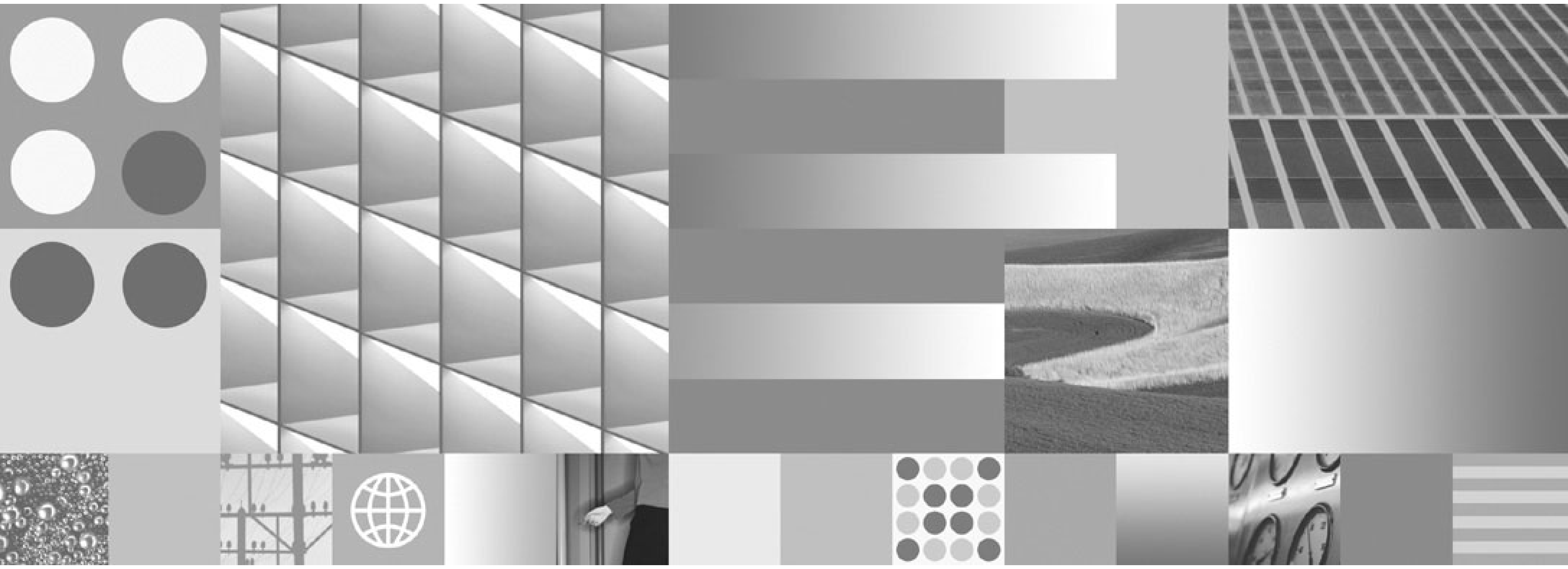
IBM DB2 9.7
for Linux, UNIX, and Windows



What's New for DB2 Version 9.7

IBM DB2 9.7
for Linux, UNIX, and Windows



What’s New for DB2 Version 9.7

SC27-2463-00

Note

Before using this information and the product it supports, read the general information under Appendix B, "Notices," on page 217.

Edition Notice

This document contains proprietary information of IBM. It is provided under a license agreement and is protected by copyright law. The information contained in this publication does not include any product warranties, and any statements provided in this manual should not be interpreted as such.

You can order IBM publications online or through your local IBM representative.

- To order publications online, go to the IBM Publications Center at www.ibm.com/shop/publications/order
- To find your local IBM representative, go to the IBM Directory of Worldwide Contacts at www.ibm.com/planetwide

To order DB2 publications from DB2 Marketing and Sales in the United States or Canada, call 1-800-IBM-4YOU (426-4968).

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

About this book

This book provides information about the new and changed functionality included in the Version 9.7 release of the DB2® Database for Linux®, UNIX®, and Windows® and DB2® Connect™ products.

Who should use this book

This book is for database administrators, application programmers, and other DB2 database users who want to find out what enhancements are available in DB2 Version 9.7 for Linux, UNIX, and Windows and in DB2 Connect Version 9.7 and what differences exist between Version 9.7 and Version 9.5 of those products.

This book provides overview information and does not contain detailed instructions for using the features described. To get additional information, use the references that are provided.

For information about the features and enhancements introduced in Version 9.7, read Part 1, "New features and functionality," on page 1.

For information about the changed, deprecated, or discontinued functionality in Version 9.7, read Part 2, "What's changed," on page 131. This information points out important changes that you need to know before using Version 9.7.

For DB2 Connect information, read Chapter 15, "DB2 Connect enhancements and changes summary," on page 127.

How this book is structured

The following topics are covered:

Part 1: New features and functionality

Chapter 1, "Product packaging enhancements," on page 3
This chapter describes the product packaging changes introduced in Version 9.7.

Chapter 2, "Compression enhancements," on page 5
This chapter describes the new features and enhancements that you can use to compress data.

Chapter 3, "Manageability enhancements," on page 9
This chapter describes the new features and enhancements that help you spend less time managing your databases.

Chapter 4, "pureXML enhancements," on page 17
This chapter describes the new pureXML® features and enhancements.

Chapter 5, "Monitoring enhancements," on page 29
This chapter describes the new features and enhancements that you can use to monitor your database systems.

Chapter 6, "High availability, backup, logging, resiliency, and recovery enhancements," on page 37
This chapter describes the new features and enhancements that help ensure that your data remains available for your users.

Chapter 7, “Performance enhancements,” on page 41

This chapter describes the new features and enhancements that help ensure the highest performance when you access and update data.

Chapter 8, “SQL compatibility enhancements,” on page 51

This chapter describes the new features and enhancements that help you port existing database applications from other vendors to DB2Version 9.7 environments.

Chapter 9, “Workload management enhancements,” on page 55

This chapter describes the new workload management features that extend the existing workload management capabilities provided in previous releases.

Chapter 10, “Security enhancements,” on page 63

This chapter describes the new features and enhancements that help you protect and manage your sensitive data.

Chapter 11, “Application development enhancements,” on page 71

This chapter describes the new features and enhancements that simplify application development, improve application portability, and ease application deployment.

Chapter 12, “SQL Procedural Language (SQL PL) enhancements,” on page 107

This chapter describes the new features and enhancements that help you work with the SQL Procedural Language (SQL PL).

Chapter 13, “Net Search Extender enhancements,” on page 117

This chapter describes the new features and enhancements for Net Search Extender.

Chapter 14, “Installation, upgrade, and fix pack enhancements,” on page 119

This chapter describes the new features and enhancements that make it faster to deploy DB2 products and easier to maintain them.

Chapter 15, “DB2 Connect enhancements and changes summary,” on page 127

This chapter describes the enhancements and changes in DB2 Version 9.7 that affect the functionality of DB2 Connect.

Part 2: What’s changed

Chapter 16, “Changed functionality,” on page 133

This chapter describes the changes to existing DB2 functionality, including changes related to database setup, database administration, application development, and CLP and system commands.

Chapter 17, “Deprecated functionality,” on page 173

This chapter lists the deprecated functionality, which refers to specific functions or features that are supported but are no longer recommended and might be removed in a future release.

Chapter 18, “Discontinued functionality,” on page 187

This chapter lists features and functionality that are unsupported in Version 9.7.

Chapter 19, “Version 9 deprecated and discontinued DB2 functionality summary,” on page 193

This chapter lists features and functionality that have been deprecated or discontinued in DB2 Version 9.1, Version 9.5, and Version 9.7.

Part 3: Appendixes

Appendix A, "Overview of the DB2 technical information," on page 205
This appendix contains information about accessing and using the latest documentation for your DB2 database systems.

Appendix B, "Notices," on page 217
This appendix contains the legal requirements and limitations related to using the DB2 database product and its documentation.

Highlighting conventions

The following highlighting conventions are used in this book.

Bold	Indicates commands, keywords, and other items whose names are predefined by the system. Commands written in uppercase are CLP commands, whereas commands written in lowercase are system commands.
Italics	Indicates one of the following: • Names or values (variables) that must be supplied by the user • General emphasis • The introduction of a new term • A reference to another source of information
`Monospace`	Indicates one of the following: • Files and directories • Information that you are instructed to type at a command prompt or in a window • Examples of specific data values • Examples of text similar to what might be displayed by the system • Examples of system messages • Samples of programming code

Part 1. New features and functionality

DB2 Version 9.7 for Linux, UNIX, and Windows delivers new capabilities that can help you manage costs and simplify application development.

Chapter 1, "Product packaging enhancements," on page 3
This chapter describes the product packaging changes introduced in Version 9.7.

Chapter 2, "Compression enhancements," on page 5
This chapter describes the new features and enhancements that you can use to compress data.

Chapter 3, "Manageability enhancements," on page 9
This chapter describes the new features and enhancements that help you spend less time managing your databases.

Chapter 4, "pureXML enhancements," on page 17
This chapter describes the new pureXML features and enhancements.

Chapter 5, "Monitoring enhancements," on page 29
This chapter describes the new features and enhancements that you can use to monitor your database systems.

Chapter 6, "High availability, backup, logging, resiliency, and recovery enhancements," on page 37
This chapter describes the new features and enhancements that help ensure that your data remains available for your users.

Chapter 7, "Performance enhancements," on page 41
This chapter describes the new features and enhancements that help ensure the highest performance when you access and update data.

Chapter 8, "SQL compatibility enhancements," on page 51
This chapter describes the new features and enhancements that help you port existing database applications from other vendors to DB2Version 9.7 environments.

Chapter 9, "Workload management enhancements," on page 55
This chapter describes the new workload management features that extend the existing workload management capabilities provided in previous releases.

Chapter 10, "Security enhancements," on page 63
This chapter describes the new features and enhancements that help you protect and manage your sensitive data.

Chapter 11, "Application development enhancements," on page 71
This chapter describes the new features and enhancements that simplify application development, improve application portability, and ease application deployment.

Chapter 12, "SQL Procedural Language (SQL PL) enhancements," on page 107
This chapter describes the new features and enhancements that help you work with the SQL Procedural Language (SQL PL).

Chapter 13, "Net Search Extender enhancements," on page 117
This chapter describes the new features and enhancements for Net Search Extender.

Chapter 14, "Installation, upgrade, and fix pack enhancements," on page 119
This chapter describes the new features and enhancements that make it faster to deploy DB2 products and easier to maintain them.

Chapter 15, "DB2 Connect enhancements and changes summary," on page 127
This chapter describes the enhancements and changes in DB2 Version 9.7 that affect the functionality of DB2 Connect.

Chapter 1. Product packaging enhancements

As IBM data servers continue to evolve, the names and the packaging of the DB2 components change to respond to market needs.

In Version 9.7, IBM has updated the list of DB2 database products available and added several new features. To read about these products and to view the related licensing and marketing information, see the DB2 Version 9 for Linux, UNIX, and Windows home page at http://www.ibm.com/db2/9.

Component names have changed

As IBM® data servers have continued to evolve, the related DB2 components and component names have changed.

The following table lists renamed product components in Version 9.7:

Table 1. New names for DB2 product components

Version 9.5 component name	Version 9.7 component name
IBM Data Server Driver for ODBC, CLI and .NET	IBM Data Server Driver Package
IBM Data Server Driver for ODBC, CLI and Open Source	IBM Data Server Driver Package

Related concepts

″IBM data server clients and drivers overview″ in Installing IBM Data Server Clients

Chapter 2. Compression enhancements

Version 9.7 includes new compression features that you can use to compress more types of data, reduce your storage requirements, improve I/O efficiency, and provide quick access to data from disk.

You can now compress the following types of data:

- XML data stored in tables (see "XML data stored in the XML storage object of tables can be compressed")
- Temporary tables (see "Temporary tables can be compressed" on page 6)
- Indexes (see "Indexes can be compressed" on page 7)
- Data replication source tables (see "Data replication source tables can be compressed" on page 7)
- Inline LOB files (see "Small LOBs can be stored in table rows and compressed" on page 48)

XML data stored in the XML storage object of tables can be compressed

XML data in the XML storage object of a table is now eligible for data row compression. In previous releases, only the table row data in a table object was eligible for compression. Data row compression saves disk space.

XML data in the XML storage object of a table is eligible for data row compression if you create the XML columns in the table in Version 9.7 and if you enable the table for data row compression. To enable a table for data row compression, use the COMPRESS YES option of the ALTER TABLE or CREATE TABLE statement.

The LOAD, REORG, and REDISTRIBUTE DATABASE PARTITION GROUP commands and the INSERT statement support compression of data in the XML storage object of a table. When the data in the XML storage object is compressed, a compression dictionary is created for the data and is stored in the XML storage object. The following table shows the effect of each command and the INSERT statement on the compression dictionary.

Table 2. Effects of commands and INSERT statement on the XML storage object compression dictionary

Directive name	Parameters	Effect on compression dictionary
LOAD command	**REPLACE** and **RESETDICTIONARY**	Creates a new compression dictionary even if one exists
	REPLACE and **KEEPDICTIONARY**	Maintains the compression dictionary if one exists; otherwise, creates a new compression dictionary
	INSERT	Creates a compression dictionary[1]
REORG command	**RESETDICTIONARY** and **LONGLOBDATA**	Creates a new compression dictionary even if one exists

Table 2. Effects of commands and INSERT statement on the XML storage object compression dictionary (continued)

Directive name	Parameters	Effect on compression dictionary
	KEEPDICTIONARY and **LONGLOBDATA**	Maintains the compression dictionary if one exists; otherwise, creates a new compression dictionary[1]
INSERT statement		Creates a compression dictionary[1]
REDISTRIBUTE DATABASE PARTITION GROUP command		Creates a compression dictionary[1]

Note: [1]Creation of a compression dictionary occurs if there is sufficient XML data in the XML storage object of the table.

Compression of data in the XML storage object of a table is not supported if the table contains XML columns from DB2 Version 9.5 or earlier. For DB2 Version 9.5 or earlier, XML columns use the type-1 XML record format. If you enable such a table for data row compression, only the table row data in the table object is compressed. To make the data in the XML storage object of the table eligible for compression, use the ADMIN_MOVE_TABLE stored procedure to migrate the table and then enable data row compression.

Related concepts

″Automatic compression dictionary creation″ in Database Administration Concepts and Configuration Reference

Related reference

″ALTER TABLE ″ in SQL Reference, Volume 2

″CREATE TABLE ″ in SQL Reference, Volume 2

Temporary tables can be compressed

With this release, temporary tables are automatically compressed, which reduces storage costs and can improve query performance.

A temporary table will be compressed under the following conditions:

- You have applied the IBM DB2 Storage Optimization Feature license.
- There is sufficient memory available at the time to build the compression dictionary.
- When executing queries, the DB2 optimizer determines that temporary table compression is worthwhile based on estimated storage savings and the impact to query performance

User temporary tables such as declared temporary tables and created temporary tables will be compressed similarly to permanent user tables as they grow.

You can use the explain facility or the db2pd tool to determine if the optimizer chose to use temporary table compression.

Related concepts

″Table compression″ in Database Administration Concepts and Configuration Reference

Indexes can be compressed

With this release, index objects can be compressed, which reduces storage costs and can improve query performance.

If data row compression is enabled on a table, indexes on the compressed tables will be compressed by default. Compression of indexes on a table can be explicitly enabled or disabled through the COMPRESS clause on the CREATE INDEX and ALTER INDEX statements.

You can also explicitly enable or disable compression for each index using the new ALTER INDEX statement, or the new COMPRESS option on the CREATE INDEX statement.

Related concepts

″Index compression″ in Database Administration Concepts and Configuration Reference

Data replication source tables can be compressed

With this release, you can enable row compression on tables that are source tables for replication. This means that the COMPRESS YES and DATA CAPTURE CHANGES options for the CREATE TABLE and ALTER TABLE statements can be specified together.

When you create or alter a table, you can specify both the DATA CAPTURE CHANGES and the COMPRESS YES clauses to have information regarding SQL changes about the table written to the log and to use data row compression. With these options enabled, as a result of REORG operations, the table can have two dictionaries: a *current data compression dictionary* and a *historical compression dictionary*.

The historical dictionary is kept (if it already exists) for data replication purposes. It is utilized whenever a log reader is delayed behind current activity, and the compression dictionary for the table or table partition has been replaced with a new dictionary using the RESETDICTIONARY option on a REORG or LOAD operation. This allows the db2ReadLog API to decompress the row contents in the log records, which were written prior to the creation of the new compression dictionary.

Note: To have log readers return the data within log records in an uncompressed format, instead of a raw compressed format, you must set the **iFilterOption** parameter of the db2ReadLog API to DB2READLOG_FILTER_ON.

Related concepts

″Table compression″ in Database Administration Concepts and Configuration Reference

Related reference

″db2ReadLog - Read log records″ in Administrative API Reference

″REORG INDEXES/TABLE ″ in Command Reference

″TRUNCATE ″ in SQL Reference, Volume 2

Chapter 3. Manageability enhancements

Version 9.7 provides enhancements that make it easier to manage DB2 environments, reduce the total cost of ownership (TCO), reduce the impact of performing system management tasks, and extend the capabilities of the autonomic features introduced in previous releases.

Version 9.7 includes the following automatic storage enhancements:

- Automatic storage support for existing databases and DMS table spaces (see "Existing databases can now use automatic storage")
- Table space rebalancing when you add or remove storage paths (see "Automatic storage supports dropping storage paths and rebalancing table spaces" on page 10)
- A new ALTER DATABASE statement option for dropping storage paths (see "Automatic storage supports dropping storage paths and rebalancing table spaces" on page 10)
- New monitoring elements for storage paths and a new administrative view, SNAPSTORAGE_PATHS (see "Automatic storage supports dropping storage paths and rebalancing table spaces" on page 10)

Unused storage extents are also easier to reclaim. You can reclaim storage from the following objects:

- DMS or automatic storage table spaces (see "Reclaiming unused space from table spaces is supported" on page 11)
- Multidimensional clustering (MDC) tables (see "Reclaiming space from multidimensional clustering (MDC) tables has been enhanced" on page 13)

The following enhancements are also included:

- Numerous new administrative views and table functions (see "Enhanced access to DB2 administration commands through SQL" on page 13)
- The ability to move table data online (see "Table data can be moved online using a new stored procedure" on page 14)
- Increased capacity for large and temporary table spaces (see "Table space capacity for large and temporary table spaces has increased" on page 14)
- Larger data warehouses support (see "Distribution map supports larger data warehouses" on page 14)
- Reduced downtime when adding database partition servers (see "Database partition servers are online immediately after being added to an instance" on page 15)
- Additional index information from the DESCRIBE command (see "DESCRIBE command provides additional index information" on page 16)

Existing databases can now use automatic storage

Existing databases that are not enabled for automatic storage can now be converted to use automatic storage. Likewise, existing DMS table spaces can now be converted to use automatic storage.

You can use the ALTER DATABASE statement to enable automatic storage for an existing database.

You can use any of the following methods to convert existing table spaces to use automatic storage:

- Convert one or more DMS table spaces by performing a redirected restore operation.
- Convert a specific DMS table space using the ALTER TABLESPACE statement.

Automatic storage simplifies table space storage management. Instead of having to manage storage at the table space level, through explicit container definitions, you can manage storage at the database level and have the DB2 data server manage the table space containers. In prior releases, automatic storage could only be used with new databases.

Related tasks

″Converting table spaces to use automatic storage″ in Database Administration Concepts and Configuration Reference

Automatic storage supports dropping storage paths and rebalancing table spaces

Starting in Version 9.7, you can begin using any storage you add to an automatic storage database immediately by rebalancing the automatic storage table spaces. You can also drop storage paths from a database managed by automatic storage.

Follow these steps to drop a storage path:

1. Drop the storage path using the ALTER DATABASE statement with the DROP STORAGE ON clause.
2. Rebalance permanent table spaces that use the storage path by using ALTER TABLESPACE with the REBALANCE clause, which moves data off the path to be dropped.
3. Drop then re-create any temporary table spaces that use the storage path that you dropped.

New monitoring elements show storage path information

New monitoring elements display information about the storage paths, including the database partition expression (for partitioned database environments only) and the current state of the path, namely not in use, in use, or drop pending. These monitor elements are available through the snapshot monitor.

The administrative views that provide information about storage paths and table space partitions have been updated.

- The SNAPSTORAGE_PATHS administrative view has been updated to display information from the SNAP_GET_STORAGE_PATHS_V97 table function instead of the SNAP_GET_STORAGE_PATHS table function.
- The SNAPTBSP_PART administrative view, which you can use to obtain information about a table space on a specific database partition, has been updated to display information from the SNAP_GET_TBSP_PART_V97 table function instead of the SNAP_GET_TBSP_PART_V91 table function.

Related concepts

″Automatic storage″ in Database Administration Concepts and Configuration Reference

Related reference

″Scenarios: Adding and removing storage with automatic storage table spaces″ in Database Administration Concepts and Configuration Reference

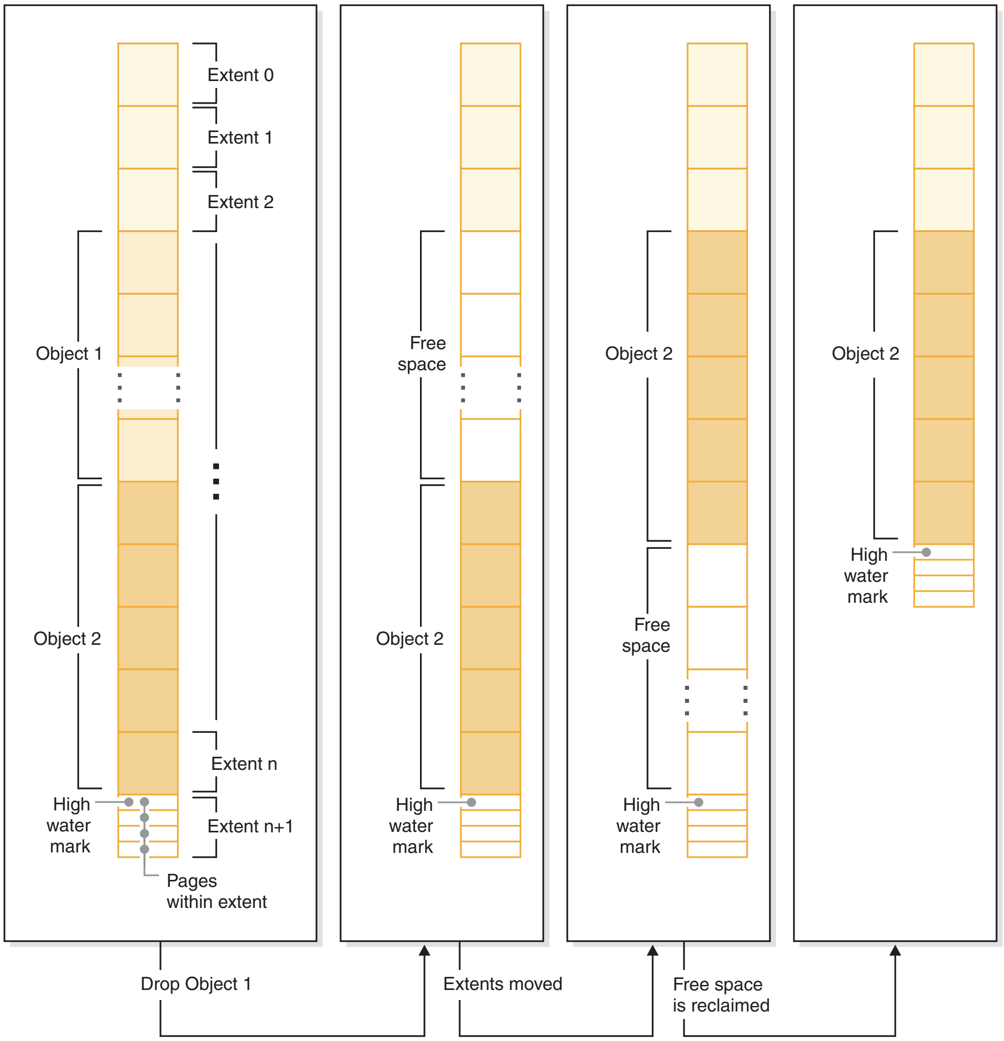
Reclaiming unused space from table spaces is supported

For a DMS or automatic storage table space created in Version 9.7, you can use *reclaimable storage* to return unused storage to the system for reuse. Reclaiming storage is an online operation; it does not impact the availability of data to users.

You can reclaim the unused storage at any time by using the ALTER TABLESPACE statement with the REDUCE option:

- For automatic storage table spaces, the REDUCE option has sub options to specify whether to reduce storage by the maximum possible amount or by a percentage of the current table space size.
- For DMS table spaces, first use the ALTER TABLESPACE statement with the LOWER HIGH WATER MARK option, and then the ALTER TABLESPACE statement with the REDUCE option and associated container operation clauses.

The storage reclamation operation remaps extents to place any unused extents towards the end of the table space. The unused space is then returned to the file system. The following diagram illustrates this process.



To take advantage of reclaimable storage with a table space created with an earlier version of the DB2 product, replace the table space with a new one created in Version 9.7. To populate the new table space, you can use one of the following methods:

- Unload and reload the data
- Move the data with an online table move operation using the ADMIN_MOVE_TABLE procedure

Table spaces for which reclaimable storage is enabled can coexist in the same database as table spaces without reclaimable storage.

You cannot reclaim storage of temporary table spaces.

Related concepts

″Reclaimable storage″ in Database Administration Concepts and Configuration Reference

Related tasks

″Moving tables online by using the ADMIN_MOVE_TABLE procedure″ in Data Movement Utilities Guide and Reference

Related reference

″ALTER TABLESPACE ″ in SQL Reference, Volume 2

Reclaiming space from multidimensional clustering (MDC) tables has been enhanced

MDC tables can be reorganized to reclaim extents that are not being used. Starting in Version 9.7, a complete offline table reorganization is no longer needed to reclaim the MDC extents.

Both the REORG TABLE command and the db2Reorg API have a new reclaim extents option. As part of this new method to reorganize MDC tables, you can also control the access to the MDC table while the reclaim operation is taking place. Your choices include: no access, read access, and write access (which is the default).

Reclaimed space from the MDC table can be used by other objects within the table space. In previous releases the free space could only be used by the MDC table.

If you use an automated maintenance policy for a database, you can update your policy file to automate reclaiming space from MDC tables. To create or update a policy file, use the AUTOMAINT_SET_POLICY procedure.

Related concepts

″Multidimensional clustering extent management″ in Partitioning and Clustering Guide

Enhanced access to DB2 administration commands through SQL

SQL administrative routines introduced in Version 9.5 are expanded in Version 9.7 to include more administrative tasks. New administrative views are also added in Version 9.7.

The SQL administrative routines and views provide a primary, easy-to-use programmatic interface to use DB2 functionality through SQL. They encompass a collection of built-in views, table functions, procedures, and scalar functions for performing a variety of administrative tasks. These routines and views can be invoked from an SQL-based application, a command line or a command script.

In addition to the new administrative views, routines, and procedures, Version 9.7 includes:

- Extended workload management capabilities.
- Expanded support for monitoring your database.
- New support for communicating through messages and alerts, and for working with files on the database server file system.
- New support for administrative routines that are data server version independent.

To provide expanded support for the existing administrative routines, some of the Version 9.5 routines have been replaced with new, more comprehensive routines or views in Version 9.7.

For the list of the new and changed routines in Version 9.7, see “Some system catalog views, system-defined administrative routines and views have been added and changed” on page 154. For the list of all supported administrative SQL routines and views, see “Supported administrative SQL routines and views” in *Administrative Routines and Views*.

Table data can be moved online using a new stored procedure

You can now call the ADMIN_MOVE_TABLE stored procedure to move the data in a table to a new table object of the same name (but with possibly different storage characteristics) while the data remains online and available for access. You can also generate a new optimal compression dictionary when a table is moved.

This feature reduces your total cost of ownership (TCO) and complexity by automating the process of moving table data to a new table object while allowing the data to remain online for select, insert, update, and delete access.

The ADMIN_MOVE_TABLE procedure creates a shadow copy of the table. During the copy phase, insert, update, and delete operations against the original table are captured using triggers and placed into a staging table. After the copy phase has completed, the data change operations that were captured in the staging table are replayed to the shadow copy. The copy of the table includes all table options, indexes, and views. The procedure then briefly takes the table offline to swap the object names.

Table space capacity for large and temporary table spaces has increased

In Version 9.7, the maximum size of large and temporary table spaces has been raised to 64 TB.

Depending on the page size you choose, the new limits on table space sizes are:

Table 3. Changes to the upper limit of table spaces based on the page size

Page size	Version 9.5 table space limit	Version 9.7 table space limit
4 KB	2 TB	8 TB
8 KB	4 TB	16 TB
16 KB	8 TB	32 TB
32 KB	16 TB	64 TB

Related reference

″SQL and XML limits″ in Database Administration Concepts and Configuration Reference

Distribution map supports larger data warehouses

In DB2 Version 9.7, the distribution map has grown from 4096 (4 KB) entries to 32 768 (32 KB) entries. This increase can greatly reduce the potential for data skew issues, thus enabling warehouses to grow far larger. To take advantage of the larger maps, set the registry variable **DB2_PMAP_COMPATIBILITY** to OFF.

A distribution map with 4096 entries can limit the size of a data warehouse because the larger the warehouse grows, the larger the potential for data skew becomes. For example, in a 10-partition database system, some database partitions appear 410 times in the distribution map, and others appear 409 times (a difference of .2%). In a 200-partition system, some database partitions appear 20 times in the distribution map, and others appear 21 times. This 5% difference of representation in the distribution map begins to indicate a significant data skew. Therefore, before DB2 Version 9.7, the largest practical size for a warehouse was approximately 200 database partition servers. With the increased size of the distribution map, the maximum skew on a system with 1000 database partition servers is 3%.

If you upgrade to DB2 Version 9.7, the size of the distribution map increases automatically. You can check the SYSCAT.PARTITIONMAPS view to see the distribution map after the upgrade. If you use the Control Center to view the SYSPARTITIONMAPS view, it shows the names of the new distribution maps.

Version 9.7 includes two new APIs (db2GetDistMap and db2GetRowPartNum) that support all distribution map sizes.

Related concepts

″Distribution maps″ in Partitioning and Clustering Guide

“sqlugtpi API has been deprecated” on page 180

“sqlugrpn API has been deprecated” on page 180

Related reference

″SYSCAT.PARTITIONMAPS ″ in Partitioning and Clustering Guide

″db2GetDistMap - Get distribution map″ in Administrative API Reference

Database partition servers are online immediately after being added to an instance

In Version 9.7, you can use the START DATABASE MANAGER command to add new database partition servers to a multi-partition database environment without having to stop and restart the instance. Thus, the database partition servers are immediately online. This enhancement reduces the cost of scaling the database as system downtime is eliminated.

As a data warehouse matures, you might have to add additional computational power to the environment to store data or support applications. As part of this process, you must add one or more new database partition servers to increase the size of the environment. Before Version 9.7, if you added a database partition server, it was not visible to the instance until after you stopped and restarted the instance. This requirement to stop and restart the instance affected system availability. Now, when you add a new database partition server, it is online after being added. When you add the new database partition server online, the following process takes place:

- The node configuration file (db2nodes.cfg) is updated automatically by the START DATABASE MANAGER command, using the values that you specify. You do not have to modify this file manually.
- The new database partition server informs the rest of the database system of its addition to the environment. New applications are aware of the new database partition server as soon as you add it. Some existing database applications are aware of the new database partition server at their transaction boundaries, and other existing applications are aware of it at their next requests.

- A skeleton database partition is created on the new database partition server for each database. If the new database partition is added to a single-partition environment, the new database partition is configured using the database configuration values of the catalog partition. If the new database partition is added to a multi-partition environment, the new database partition is configured using database configuration values from a non-catalog database partition. If a problem occurs during the database partition configuration, the new database partition is configured using default database configuration parameter values.

You can monitor the progress of the add database partition server operation by using the **-addnode** parameter of the db2pd command.

If you want to use the START DATABASE MANAGER command to add a new database partition server to the environment but do not want it to be active after adding it, you can set the **DB2_FORCE_OFFLINE_ADD_PARTITION** registry variable to TRUE and restart the instance after processing of the addition is completed.

Related reference

″START DATABASE MANAGER ″ in Command Reference

″Partitioned database environment variables″ in Partitioning and Clustering Guide

″db2pd - Monitor and troubleshoot DB2 database ″ in Command Reference

DESCRIBE command provides additional index information

The DESCRIBE command with the **INDEXES FOR TABLE** parameter now provides information about the system-generated XML regions index and XML path indexes, and DB2 Text Search indexes, in addition to information about relational indexes and indexes over XML data.

If you specify the **INDEXES FOR TABLE** parameter with the SHOW DETAIL clause, the DESCRIBE command lists more information for all the types of indexes. The **INDEXES FOR TABLE** parameter also supports the new options RELATIONAL DATA, XML DATA, and TEXT SEARCH to list information about a specific type of index.

You might be able to improve performance by using the DESCRIBE command to list the indexes for a table and evaluating whether to add new indexes or drop unused indexes.

Related reference

″DESCRIBE ″ in Command Reference

Chapter 4. pureXML enhancements

Version 9.7 builds on the pureXML support first introduced in Version 9.1 and enhances the hybrid relational and XML data server to make your XML data processing even more flexible, faster, and more reliable. These enhancements provide new opportunities to deploy and analyze XML data in data warehouses.

In Version 9.7, XML data is supported in the following additional types of tables, objects, and environments:

- Partitioned tables (see "Partitioned tables support XML data" on page 18)
- Multidimensional clustering (MDC) tables (see "MDC tables support XML columns" on page 18)
- Declared temporary tables (see "Declared temporary tables support XML columns" on page 19)
- User-defined functions (see "Inlined SQL functions support the XML data type" on page 20)
- Partitioned database environments (see "Partitioned database environments support the pureXML feature" on page 21)

The following other enhancements extend pureXML support:

- The DESCRIBE command provides information about system-generated XML indexes (see "DESCRIBE command provides additional index information" on page 16).
- The ALTER TABLE statement supports REORG-recommended operations containing XML data (see "The ALTER TABLE statement supports REORG-recommended operations containing XML data" on page 22).
- Predicate pushdown is available for SQL/XML and XQuery statements (see "Predicate pushdown is available for SQL/XML and XQuery statements" on page 22).
- XML parsing and validation can return more detailed messages (see "XML parsing and validation can return more detailed messages" on page 22.)
- New functions display base table row storage information for inlined XML documents (see "Small LOBs can be stored in table rows and compressed" on page 48).

pureXML performance has been enhanced as follows:

- Multiple XML documents in a column can be decomposed at the same time (see "Multiple XML documents in a column can be decomposed at the same time" on page 23).
- Optimization profiles support guidelines for XML data (see "Optimization profiles support guidelines for XML data" on page 24).
- Write access is supported while you are creating or reorganizing indexes over XML data (see "Write access is supported during creation or reorganization of indexes over XML data" on page 26).

You can use another enhancement to compress XML data. For more information, see "XML data stored in the XML storage object of tables can be compressed" on page 5.

You can use new sample programs to learn about pureXML enhancements.

Related concepts

″pureXML overview -- DB2 as an XML database″ in pureXML Guide

″pureXML tutorial″ in pureXML Guide

″XML input and output overview″ in pureXML Guide

“New DB2 sample programs have been added” on page 89

Partitioned tables support XML data

Starting with DB2 Version 9.7, partitioned tables can include XML data and can benefit from the easy roll-in and roll-out of data provided by the table partitioning functionality. In addition to queries over relational data, queries over XML data can also benefit from the performance advantage of partition elimination.

Partitioned tables use a data organization scheme in which table data is divided among multiple storage objects, called data partitions, according to values in one or more table-partitioning key columns of the table. A partitioned table simplifies the rolling in and rolling out of table data.

You can use partitioned tables with the pureXML feature as follows:

- You can create a partitioned table with one or more XML columns by using the CREATE TABLE statement.
- You can add an XML column to an existing partitioned table by using the ALTER TABLE statement with the ADD COLUMN clause.
- You can modify a partitioned table containing XML data by using the ALTER TABLE statement with the ADD PARTITION, ATTACH PARTITION, and DETACH PARTITION clauses.
- You can place each data partition and its associate XML storage object (XDA) into the same or different table spaces.
- You can place each nonpartitioned index over XML data in a different table space and reorganize each index separately.
- With effective storage planning, you can individually back up any partitioned data or nonpartitioned indexes by performing table space backups.

You can use new samples program to learn how to use some of these features.

Note: You cannot use XML-type columns as table-partitioning key columns.

Related concepts

″Partitioned tables″ in Partitioning and Clustering Guide

″Logical and physical indexes over XML data″ in pureXML Guide

“New DB2 sample programs have been added” on page 89

Related tasks

″Migrating existing tables and views to partitioned tables″ in Partitioning and Clustering Guide

Related reference

″Restrictions on the pureXML feature″ in pureXML Guide

MDC tables support XML columns

Multidimensional clustering (MDC) tables can now be created with XML columns, and XML columns can now be added to MDC tables.

MDC provides a method for clustering data in tables along multiple dimensions. MDC tables can significantly improve query performance and reduce the overhead of data maintenance operations such as reorganizing data, inserting data, and deleting data.

You can now create MDC tables that contain one or more XML columns, add XML columns to MDC tables using the ADD COLUMN clause of the ALTER TABLE statement, and create indexes over XML data in MDC tables. Queries can use both indexes over XML data and MDC indexes to enhance performance.

You cannot specify an XML column as a dimension in the ORGANIZE BY clause of the CREATE TABLE statement.

You can use new sample programs to learn how to use some of these features.

Example

In the following example, an MDC table called CUST_INFO is created. The MDC table contains a column of type XML and uses the columns REGION, AGE, and INCOME as dimensions:

```
CREATE TABLE cust_info(id BIGINT, age INT, region VARCHAR(10), income DECIMAL(10,2), history XML)
     ORGANIZE BY (region, age, income);
```

Related concepts

″Multidimensional clustering tables″ in Partitioning and Clustering Guide

″Table and index management for MDC tables″ in Troubleshooting and Tuning Database Performance

″pureXML overview -- DB2 as an XML database″ in pureXML Guide

″Restrictions on indexes over XML data″ in pureXML Guide

“New DB2 sample programs have been added” on page 89

Related reference

″Restrictions on the pureXML feature″ in pureXML Guide

Declared temporary tables support XML columns

In DB2 Version 9.7, declared temporary tables can contain XML columns.

Prior to DB2 Version 9.7, you could not store XML data in declared temporary tables, so applications either had to work around not having the XML data in a declared temporary table or had to use a regular table.

You can use a new sample program to learn how to use this feature.

The following functionality is not supported for declared temporary tables, either with XML data or relational data:

- Data redistribution
- Table partitioning
- Multidimensional clustering

In addition, you cannot do the following with declared temporary tables, either with XML data or relational data:

- Specify declared temporary tables in an ALTER, COMMENT, GRANT, LOCK, RENAME or REVOKE statement.

- Reference declared temporary tables in a CREATE ALIAS, CREATE FUNCTION (SQL Scalar, Table, or Row), CREATE TRIGGER, or CREATE VIEW statement.
- Specify declared temporary tables in referential constraints.
- Use the LOAD or IMPORT command to add data to declared temporary tables.
- Use the REORG command to reorganize data or indexes of declared temporary tables.

Related concepts

“New DB2 sample programs have been added” on page 89

″Using declared temporary tables with XML data″ in pureXML Guide

Related reference

″DECLARE GLOBAL TEMPORARY TABLE ″ in SQL Reference, Volume 2

Inlined SQL functions support the XML data type

The XML data type is now supported for inlined SQL functions that you create using the CREATE FUNCTION (SQL scalar, table, or row) statement or CREATE FUNCTION (sourced or template) statement.

Use the CREATE FUNCTION (SQL scalar, table, or row) statement to define an user-defined SQL scalar, table, or row function, and use the CREATE FUNCTION (sourced or template) statement to register a user-defined function based on a user-defined SQL scalar function.

A user-defined function created with CREATE FUNCTION (SQL scalar, table or row) can use input, output, or input/output parameters of type XML. You can use XML variables in SQL statements in the same way as variables of any other data type. For example, in a user-defined function you can pass variables of data type XML as parameters to XQuery expressions in an XMLEXISTS predicate or a function like XMLQUERY or XMLTABLE.

In a user-defined function created with CREATE FUNCTION (sourced or template) that invokes a user-defined SQL scalar function as the source function, you can use input, output, or input/output parameters of type XML.

XML values are assigned by reference in a user-defined function.

Parameters and variables of data type XML are not supported in compiled SQL functions.

You can use new sample programs to learn how to use some of these features.

Related concepts

“New DB2 sample programs have been added” on page 89

″Inlined SQL functions and compiled SQL functions″ in pureXML Guide

“XML data is passed by reference in SQL stored procedures” on page 166

Related reference

″CREATE FUNCTION (SQL scalar, table, or row) ″ in SQL Reference, Volume 2

″CREATE FUNCTION (Sourced or Template) ″ in SQL Reference, Volume 2

Partitioned database environments support the pureXML feature

In partitioned database environments, tables containing XML columns can be stored in multi-partition databases. Starting in Version 9.7, XML data in these environments can also be managed using the DB2 pureXML feature.

The pureXML feature allows you to store, query, and manage well-formed XML documents in table columns that have the XML data type to leverage DB2 data server functionality and to deploy new business applications.

Partitioned database environments allows tables to be distributed across multi-partition databases to use the power of multiple processors on multiple machines to improve query performance.

In DB2 Version 9.7, the pureXML feature is supported in partitioned database environments. With both features tightly integrated, pureXML customers can distribute XML data across multiple database partitions and parallelize XML queries for better performance, while partitioned database environments customers can deploy pureXML for new business applications.

You can use the following pureXML functionality in partitioned database environments:

- Create a table that uses a distribution key and that contains XML columns by connecting to any database partition.
- Create XML value indexes by connecting to any partition.
- Register, add, complete, alter, and drop an XML schema, DTD, or external entity as an XML schema repository (XSR) object by connecting to any partition. Use registered and completed XSR objects to validate or decompose XML documents, even when the XSR objects and the XML documents reside on different partitions.
- Use SQL and SQL/XML functions to query, insert, update, delete, or publish XML data. The data operations are parallelized as much as possible, based on the partitioning of the XML data.
- Use the existing XML constraint and trigger support.
- Use the XQuery programming language to query data residing on multiple partitions.
- Load large volumes of XML data in parallel into tables that are distributed across database partitions.
- Use the LOAD command with the ANYORDER file type modifier when loading XML data into an XML column. ANYORDER is also supported in a single-partition database environment.
- Issue the RUNSTATS command against a table with XML columns by connecting to any partition.
- Use the XQuery transform functionality.

- Store XML documents inline in the rows of the base table instead of storing them in the default XML storage object.
- Use the Visual Explain tool to identify a new type of table queue operator (XTQ) that is displayed for access plans generated for XQuery in partitioned databases.

You can use new sample programs to learn how to use some of these features.

Related concepts

″Partitioned database environments″ in Partitioning and Clustering Guide

″pureXML overview -- DB2 as an XML database″ in pureXML Guide

″XML schema repository″ in pureXML Guide

″SQL/XML publishing functions for constructing XML values″ in pureXML Guide

″Loading XML data″ in pureXML Guide

“New DB2 sample programs have been added” on page 89

Related reference

″CREATE INDEX ″ in SQL Reference, Volume 2

″RUNSTATS ″ in Command Reference

″Restrictions on the pureXML feature″ in pureXML Guide

Related information

″Introduction to XQuery″ in pureXML Guide

The ALTER TABLE statement supports REORG-recommended operations containing XML data

Prior to DB2 Version 9.7, for any tables containing columns of data type XML, a REORG-recommended operation could not be specified in an ALTER TABLE statement. This restriction has been lifted.

Using the ALTER TABLE command, you can specify any REORG-recommended operation for any table containing columns of type XML. However, an operation that drops XML columns must drop all of the XML columns in the table in a single ALTER TABLE statement.

Related reference

″ALTER TABLE ″ in SQL Reference, Volume 2

Predicate pushdown is available for SQL/XML and XQuery statements

The DB2 optimizer uses the predicate pushdown query optimization technique which enables early data filtering and better potential index usage. Starting in Version 9.7, the compiler pushes down predicates (for filters and XPath extractions) into XQuery query blocks. This technique is similar to the predicate pushdown optimization technique for SQL statements.

Related concepts

″Compiler rewrite example: Predicate pushdown for combined SQL/XQuery statements″ in Troubleshooting and Tuning Database Performance

XML parsing and validation can return more detailed messages

Version 9.7 includes the stored procedure XSR_GET_PARSING_DIAGNOSTICS, which returns detailed error messages during XML parsing and validation.

The XSR_GET_PARSING_DIAGNOSTICS stored procedure improves error handling as follows:

- It can report multiple errors at the same time.
- It indicates the location of an error in a document in two ways: as a column and line number, and as an XPath.
- It provides the original XML4C error along with the DB2 SQLCODE and reason code.
- It returns all of the information in XML format.

You can provide the XML schema used for validation in the following ways:

- Through the name registered in the XML schema repository (XSR)
- Through the schema URL
- Implicitly, through the XML document itself

The XSR_GET_PARSING_DIAGNOSTICS stored procedure was introduced in DB2 Version 9.5 Fix Pack 3.

Related concepts

″Displaying detailed XML parsing and validation errors″ in pureXML Guide

Related reference

″ErrorLog XML schema definition for enhanced error message support″ in pureXML Guide

″XSR_GET_PARSING_DIAGNOSTICS stored procedure″ in pureXML Guide

Multiple XML documents in a column can be decomposed at the same time

You can use the new CLP command DECOMPOSE XML DOCUMENTS to decompose multiple XML documents stored in a binary or an XML column. The command stores data from the XML documents in the columns of one or more relational tables, based on annotations specified in a registered annotated XML schema.

Annotated XML schema decomposition is one way to store pieces of an XML document in columns of one or more tables. This type of decomposition breaks an XML document down for storage in tables, based on the annotations specified in a registered annotated XML schema. See the related links for information on annotated XML schema decomposition.

You can use the corresponding stored procedure XDB_DECOMP_XML_FROM_QUERY to decompose one or more XML documents stored in a binary column or an XML column.

The new CLP command and stored procedure are extensions of the DB2 annotated XML schema decomposition feature that decomposes a single XML document.

Related concepts

″Annotated XML schema decomposition″ in pureXML Guide

Related reference

″XDB_DECOMP_XML_FROM_QUERY stored procedure for annotated schema decomposition″ in pureXML Guide

″DECOMPOSE XML DOCUMENTS ″ in Command Reference

Optimization profiles support guidelines for XML data

You can add optimization guidelines to optimization profiles to maximize the performance of queries used in database applications. Version 9.7 supports new XML-specific optimization guidelines, as well as the use of existing optimization guidelines to influence the access plans of queries against XML data.

For queries that access XML data or that use indexes over XML data, you can specify the following types of optimizations in an optimization guideline:

- Control how XML data is moved between partitions in a partitioned database environment by using the DPFXMLMOVEMENT general request element.
- Control the join order for joins on XML data types in plan optimization guidelines by setting the attribute FIRST=″TRUE″ in access request elements or by using join request elements.
- Control usage of indexes over XML data by using one of the following options:
  - Use the XISCAN access request element to request the optimizer choose a single XML index scan to access a table.
  - Use the XANDOR access request element to request the optimizer choose multiple XANDORed XML index scans to access a table.
  - Use the IXAND access request element with the TYPE attribute value set to XMLINDEX to request the optimizer choose multiple relational and XML index scans.
  - Use the ACCESS access request element and specify the attribute TYPE=″XMLINDEX″ to request the optimizer use a cost-based analysis to choose any one of the available XML index access techniques to access a table.
  - Use the ACCESS access request element and specify the attributes TYPE=″XMLINDEX″ and ALLINDEXES=″TRUE″ to request the optimizer use all of the applicable relational indexes and indexes over XML data to access the specified table, regardless of cost.
  - Use the IXAND access request element and specify the attributes TYPE=″XMLINDEX″ and ALLINDEXES=″TRUE″ to request the optimizer use all applicable relational indexes and indexes over XML data in an IXAND plan to access the specified table, regardless of cost.

Related concepts

″Index ANDing access requests″ in Troubleshooting and Tuning Database Performance

″DPFXMLMOVEMENT requests″ in Troubleshooting and Tuning Database Performance

″XML index scan access requests″ in Troubleshooting and Tuning Database Performance

″XML index ANDing and ORing access requests″ in Troubleshooting and Tuning Database Performance

″Using optimization guidelines with XML data and XQuery expressions″ in pureXML Guide

″Examples of optimization guidelines with XML data″ in pureXML Guide

XML data stored in the XML storage object of tables can be compressed

XML data in the XML storage object of a table is now eligible for data row compression. In previous releases, only the table row data in a table object was eligible for compression. Data row compression saves disk space.

XML data in the XML storage object of a table is eligible for data row compression if you create the XML columns in the table in Version 9.7 and if you enable the table for data row compression. To enable a table for data row compression, use the COMPRESS YES option of the ALTER TABLE or CREATE TABLE statement.

The LOAD, REORG, and REDISTRIBUTE DATABASE PARTITION GROUP commands and the INSERT statement support compression of data in the XML storage object of a table. When the data in the XML storage object is compressed, a compression dictionary is created for the data and is stored in the XML storage object. The following table shows the effect of each command and the INSERT statement on the compression dictionary.

Table 4. Effects of commands and INSERT statement on the XML storage object compression dictionary

Directive name	Parameters	Effect on compression dictionary
LOAD command	**REPLACE** and **RESETDICTIONARY**	Creates a new compression dictionary even if one exists
	REPLACE and **KEEPDICTIONARY**	Maintains the compression dictionary if one exists; otherwise, creates a new compression dictionary
	INSERT	Creates a compression dictionary[1]
REORG command	**RESETDICTIONARY** and **LONGLOBDATA**	Creates a new compression dictionary even if one exists
	KEEPDICTIONARY and **LONGLOBDATA**	Maintains the compression dictionary if one exists; otherwise, creates a new compression dictionary[1]
INSERT statement		Creates a compression dictionary[1]

Table 4. Effects of commands and INSERT statement on the XML storage object compression dictionary (continued)

Directive name	Parameters	Effect on compression dictionary
REDISTRIBUTE DATABASE PARTITION GROUP command		Creates a compression dictionary[1]

Note: [1]Creation of a compression dictionary occurs if there is sufficient XML data in the XML storage object of the table.

Compression of data in the XML storage object of a table is not supported if the table contains XML columns from DB2 Version 9.5 or earlier. For DB2 Version 9.5 or earlier, XML columns use the type-1 XML record format. If you enable such a table for data row compression, only the table row data in the table object is compressed. To make the data in the XML storage object of the table eligible for compression, use the ADMIN_MOVE_TABLE stored procedure to migrate the table and then enable data row compression.

Related concepts

″Automatic compression dictionary creation″ in Database Administration Concepts and Configuration Reference

Related reference

″ALTER TABLE ″ in SQL Reference, Volume 2

″CREATE TABLE ″ in SQL Reference, Volume 2

Write access is supported during creation or reorganization of indexes over XML data

Transactions can now insert, update, and delete data in a table during creation or reorganization of indexes over XML data on the table when the table is a nonpartitioned table or a partitioned table.

Because transactions that insert, update, or delete data in a table no longer have to wait for the creation or reorganization of indexes on that table to be completed, throughput for concurrent transactions is higher, and response time for concurrent transactions is faster.

In Version 9.7, the following indexing operations support concurrent read and write access to a table:

- For a nonpartitioned table:
  - Using the CREATE INDEX statement on an XML column
  - Using the REORG INDEXES ALL FOR TABLE command with the ALLOW WRITE ACCESS option on a table with one or more XML columns
- For a partitioned table:
  - Using the CREATE INDEX statement to create a nonpartitioned index over XML data
  - Using the REORG INDEX command with the ALLOW WRITE ACCESS option on a nonpartitioned index over XML data

You can use a new sample program to learn how to use this feature.

Related concepts

″Indexing XML data″ in pureXML Guide

“New DB2 sample programs have been added” on page 89

Related reference

″CREATE INDEX ″ in SQL Reference, Volume 2

″REORG INDEXES/TABLE ″ in Command Reference

Chapter 5. Monitoring enhancements

Version 9.7 includes many enhancements that make monitoring DB2 environments more comprehensive with higher granularity of control.

DB2 Version 9.7 accelerates problem solving through new point in time and event monitoring. You can now better understand what is happening inside your DB2 data server by using the new comprehensive monitoring information designed to address common diagnostic problems. Since the new monitoring is also more efficient, this new level of understanding does not have a high impact on performance.

Version 9.7 includes a new monitoring infrastructure that you can access through new table functions and new event monitors. This infrastructure is a superior alternative to the existing system monitor, event monitors, snapshot commands, and snapshot SQL interfaces. This infrastructure provides the following benefits:

- New monitoring interfaces accessible through SQL (see "New relational monitoring interfaces are light weight and SQL accessible")
- A high granularity of control over what information to collect (see "New monitor elements and database configuration parameters provide more granular monitoring" on page 30)
- Ability to monitor both static and dynamic SQL statements stored in the package cache (see "New relational interface to monitor dynamic and static SQL statements in package cache" on page 32)
- New lock event monitors to simplify problem determination for deadlocks, lock timeout, and lock waits (see "Lock event reporting has been enhanced" on page 34)
- Improved transaction monitoring of unit of work and total CPU usage (see "A new unit of work event monitor supports transaction monitoring" on page 32)

The monitoring enhancements provide new monitoring information to address performance problems and other situations:

- A more comprehensive set of monitor elements indicating where and how the DB2 database manager spends its waiting time (see "Time-spent monitor elements are more comprehensive" on page 33)
- Better reporting of information about statements in the package cache, including static and dynamic statements (see "New relational monitoring interfaces are light weight and SQL accessible")
- Collection of the section details for SQL activity events (see "The section for SQL statements captured by activity event monitor can now be collected" on page 35)

New relational monitoring interfaces are light weight and SQL accessible

DB2 Version 9.7 provides new relational monitoring interfaces, that can be accessed directly by SQL, resulting in enhanced reporting and monitoring of the database system, data objects, and the package cache to help you quickly identify issues that might be causing problems.

The new interfaces report monitoring elements that provide information about work done on the system, data objects such as tables, indexes, buffer pools, table spaces and containers, and SQL entries in the package cache. The new interfaces, just like the workload management (WLM) table functions created for DB2 Version 9.5, are more efficient and have a lower impact on the system than existing system monitor and snapshot interfaces.

System, activity, and data object level monitoring information can be accessed directly through SQL using the following table functions:

System level

- MON_GET_CONNECTION
- MON_GET_CONNECTION_DETAILS
- MON_GET_SERVICE_SUBCLASS
- MON_GET_SERVICE_SUBCLASS_DETAILS
- MON_GET_UNIT_OF_WORK
- MON_GET_UNIT_OF_WORK_DETAILS
- MON_GET_WORKLOAD
- MON_GET_WORKLOAD_DETAILS

Activity level

- MON_GET_ACTIVITY_DETAILS
- MON_GET_PKG_CACHE_STMT

Data object level

- MON_GET_BUFFERPOOL
- MON_GET_CONTAINER
- MON_GET_EXTENT_MOVEMENT_STATUS
- MON_GET_INDEX
- MON_GET_TABLE
- MON_GET_TABLESPACE

New monitor elements and database configuration parameters provide more granular monitoring

DB2 Version 9.7 provides new monitor elements that enable you to perform more granular monitoring, without using the monitor switches or snapshot interfaces. Database-wide monitoring control is provided by new database configuration parameters.

With the new monitor elements and infrastructure, you can use SQL statements to efficiently collect monitor data to determine whether specific aspects of the system are working correctly and to help you diagnose performance problems, while incurring a reasonable performance overhead. With the new access methods, you can get all the data you need without using the snapshot interfaces. The increased monitoring granularity gives you more control over the data collection process; collect the data you want from the source you want.

Monitoring information is collected about the work performed by your applications and reported through table function interfaces at the following three levels:

System level

These monitoring elements provide details about all work being performed on the system. Monitor-element access points include service subclass, workload definition, unit of work, and connection.

Activity level
These monitor elements provide details about activities being performed on the system (a specific subset of the work being performed on the system). You can use these elements to understand the behavior and performance of activities. Monitor-element access points include individual activities, and entries in the database package cache.

Data object level
These monitoring elements provide details about the work being processed by the database system within specific database objects such as indexes, tables, buffer pools, table spaces, and containers, thereby enabling you to quickly identify issues with particular data objects that might be causing system problems. Monitor-element access points include buffer pool, container, index, table, and table space.

For the list of the table functions in each level, see “New relational monitoring interfaces are light weight and SQL accessible” on page 29.

For database-wide control over the collection of monitoring data at the system, activity and data object levels, and the generation of events in unit of work and locking event monitors, eight new configuration parameters have been added. The default settings are intended to provide the minimum level of collection and event generation that will be enabled for work running in all DB2 workloads and service classes. Monitoring control can be further customized by modifying DB2 workload and service class definitions. For example, you could disable collection of system level metrics for the entire database and enable collection for a particular service class if you are only interested in monitoring work performed in that service class.

Table 5. Monitor collection database configuration parameters

Parameter name	Description	Details
mon_act_metrics	Monitoring activity metrics	Controls collection of activity level monitor elements on the entire database. This will affect all DB2 workload definitions.
mon_deadlock	Monitoring deadlock	Controls the generation of deadlock events at the database level for the lock event monitor.
mon_locktimeout	Monitoring lock timeout	Controls the generation of lock timeout events at the database level for the lock event monitor. This will affect all DB2 workload definitions.
mon_lockwait	Monitoring lock wait	Controls the generation of lock wait events at the database level for the lock event monitor.
mon_lw_thresh	Monitoring lock wait threshold	The amount of time spent in lock wait (specified in microseconds) before an event for **mon_lockwait** is generated.
mon_obj_metrics	Monitoring object metrics	Controls collection of data object monitor elements on the entire database.
mon_req_metrics	Monitoring request metric	Controls collection of request monitor elements on the entire database. This will affect all DB2 service classes.

Table 5. Monitor collection database configuration parameters (continued)

Parameter name	Description	Details
mon_uow_data	Monitoring unit of work events	Controls the generation of unit of work events at the database level for the unit of work event monitor. This will affect all DB2 workload definitions.

Related concepts

″Monitor elements reported in monitor table functions″ in Database Monitoring Guide and Reference

New relational interface to monitor dynamic and static SQL statements in package cache

DB2 Version 9.7 provides a new relational interface, MON_GET_PKG_CACHE_STMT, to monitor dynamic and static SQL statements in the database package cache. This new relational interface reports information for both static and dynamic SQL statements, unlike the dynamic SQL snapshot which only reports information for dynamic statements.

For each dynamic and static SQL statement, the new relational interface returns a rich set of metrics, aggregated across executions of the statement. The metrics can help you to quickly determine the reasons for poor performance of an SQL statement, to compare the behavior and performance of one SQL statement to another, and to easily identify the most expensive SQL statements along any number of dimensions (for example, the SQL statements consuming the most CPU resources, and statements with the longest lock wait times).

A new unit of work event monitor supports transaction monitoring

The new unit of work event monitor (CREATE EVENT MONITOR FOR UNIT OF WORK) is an enhanced replacement to the deprecated transaction event monitor (CREATE EVENT MONITOR FOR TRANSACTIONS). The new unit of work event monitor contains many additional monitor elements and is more efficient than the transaction event monitor.

A common use for the new unit of work event monitor would be, as a data server provider, to determine how much to charge application users based on the amount of resources used by the application. In such billing circumstances, total CPU usage is the most commonly used resource upon which to base chargeback billing. Total CPU usage is one of the monitor elements for which data is collected in the new unit of work event monitor.

The core data collected for a unit of work event are the monitor elements reported through the MON_GET_UNIT_OF_WORK and MON_GET_UNIT_OF_WORK_DETAILS table functions. This data is enriched with a variety of information, including attributes at the database level, connection level, and unit of work level.

You can also choose to access the event monitor data through an XML document by using the new EVMON_FORMAT_UE_TO_XML table function, or through a relational table by using the new EVMON_FORMAT_UE_TO_TABLES procedure.

After using the unit of work event monitor to capture binary unit of work event data in an unformatted event table, you can convert the data into an XML or text document by using the new Java™-based db2evmonfmt tool.

Related concepts

″db2evmonfmt tool for reading event monitor data″ in Database Monitoring Guide and Reference

Related tasks

″Collecting unit of work event data and generating reports″ in Database Monitoring Guide and Reference

Related reference

″CREATE EVENT MONITOR ″ in SQL Reference, Volume 2

″CREATE EVENT MONITOR (unit of work) ″ in SQL Reference, Volume 2

″mon_req_metrics - Monitoring request metrics configuration parameter″ in Database Administration Concepts and Configuration Reference

″mon_uow_data - Monitoring unit of work events configuration parameter″ in Database Administration Concepts and Configuration Reference

Time-spent monitor elements are more comprehensive

In Version 9.7, you can use a more comprehensive set of time-based monitor elements to understand where and how the DB2 database manager spends its time. With the ability to pinpoint where most of the time is spent, you can locate potential sources of problems more easily and determine whether tuning can be done to improve performance.

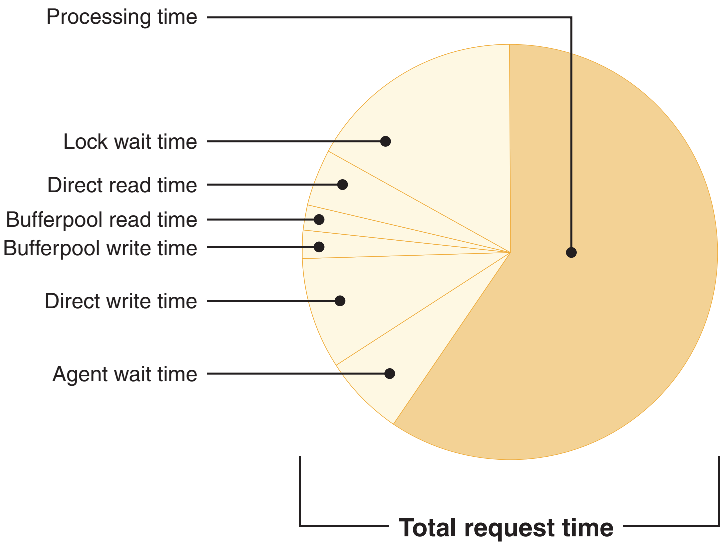
The time-spent monitor elements included in Version 9.7 provide the following information:

- Total time spent processing requests and total wait time within the DB2 database manager. Use this to approximate system utilization; as well as how much time the database manager spends working actively on requests versus waiting on a resource.
- Detailed breakdown of wait times by resource (such as lock, buffer pool, or logging). This breakdown allows you to identify the primary contributors to wait time within the DB2 database manager.
- Measurement of time spent outside the DB2 database manager (client_idle_wait_time). This allows you to identify whether a slowdown in performance is occurring inside or outside the DB2 database manager.

The time-spent monitor elements complement other time-based types of monitor elements, such as CPU time (the amount of CPU used) provided by the operating system, and monitor elements that approximate overall application response time provided by the DB2 database manager.

Example

The total request time on a particular system is spent as follows:



In this example, the new wait time metrics show that lock waits are responsible for a good portion of the total wait time. You can then take the necessary steps to further analyze the cause of these lock waits.

Note: A new Version 9.7 feature lets you collect additional information about lock events. For more information, see "Lock event reporting has been enhanced."

Lock event reporting has been enhanced

In Version 9.7, there are new approaches for collecting lock event reports, using data captured by the new lock event monitor (CREATE EVENT MONITOR FOR LOCKING). You can use this data to help identify and resolve lock wait, lock timeout, and deadlock problems.

The new lock event monitor contains many additional monitor elements and is a more efficient method to collect lock-related events than the previous methods. The new approach collects information about lock timeouts, deadlocks, and lock waits that are more than a specified duration. The previous methods included using the DB2DETAILDEADLOCK event monitor, the DB2_CAPTURE_LOCKTIMEOUT registry variable, and the CREATE EVENT MONITOR FOR DEADLOCKS statement, all of which are now deprecated.

After using the lock event monitor to capture binary lock event data in an unformatted event table, you can convert the data into an XML or text document by using the new Java-based db2evmonfmt tool.

You can also choose to access the event monitor data through an XML document by using the new EVMON_FORMAT_UE_TO_XML table function, or through a relational table by using the new EVMON_FORMAT_UE_TO_TABLES procedure.

Related concepts

″db2evmonfmt tool for reading event monitor data″ in Database Monitoring Guide and Reference

″Monitoring database locking″ in Database Monitoring Guide and Reference

″Diagnosing and resolving locking problems″ in

Related tasks

″Collecting lock event data and generating reports″ in Database Monitoring Guide and Reference

Related reference

″CREATE EVENT MONITOR (locking) ″ in SQL Reference, Volume 2

The section for SQL statements captured by activity event monitor can now be collected

The section for an SQL statement can now be collected by an activity event monitor, along with the rest of the activity information, by specifying the COLLECT ACTIVITY DATA WITH DETAILS,SECTION clause on a service class, workload, work action set, or threshold. An explanation of the used access plan can be extracted from the section using the db2expln command.

The granular control in collecting a section for an SQL statement allows you to limit the amount of data collected, gathering only the data of interest for a particular problem determination task. For example, use a threshold to only collect a section for statements that run for a particularly long time or exceed a certain cost.

Related tasks

″Collecting data for individual activities″ in Workload Manager Guide and Reference

Related reference

″db2expln - SQL and XQuery Explain ″ in Command Reference

Chapter 6. High availability, backup, logging, resiliency, and recovery enhancements

Version 9.7 includes enhancements that help ensure that your data remains available.

You can use the existing integrated high availability features to configure clustered environments on Solaris SPARC. For more information, see "Cluster management software integration support has been extended (Solaris)."

Improved DB2 infrastructure provides the following benefits:

- Enhanced resilience to errors and traps caused by certain categories of critical errors (see "Enhanced resilience to errors and traps reduces outages").
- Greater control over the maximum size of administration notification and diagnostic logs (see "Administration notification and diagnostic logs occupy specified amount of disk space" on page 38).

Cluster management software integration support has been extended (Solaris)

In Version 9.7, you can use the existing integrated high availability features to configure clustered environments on Solaris SPARC, in addition to the existing support on the AIX® and Linux operating systems.

The DB2 cluster manager application programming interface (API) enables you to use IBM Data Server cluster configuration tools such as the DB2 high availability instance configuration utility (db2haicu) to configure clustered environments.

An updated version of the IBM Tivoli® System Automation for Multiplatforms (SA MP) Base Component is integrated with IBM Data Server product installations as part of the DB2 High Availability Feature on Solaris SPARC, Linux, and AIX operating systems.

Related tasks

″Configuring a clustered environment using DB2 High Availability Instance Configuration Utility (db2haicu)″ in Data Recovery and High Availability Guide and Reference

Enhanced resilience to errors and traps reduces outages

Improvements in the Version 9.7 infrastructure enhance its resilience to certain categories of critical errors and traps. For example, a DB2 instance can remain operational when certain critical data errors are encountered when reading from disk or in-memory data pages, as well as when unexpected traps occur.

In Version 9.7, there are an increased number of cases in which a DB2 instance continues to safely operate even when critical errors and traps occur, and all the while providing diagnostic messages that help in tracking these events. The following additional messages can be generated in the administration notification log and should be monitored by database administrators:

- ADM6006E
- ADM6007C

- ADM14011C
- ADM14012C
- ADM14013C

In addition, the following SQLCODEs can be received by applications:
- SQL1655E
- SQL1656E

Remaining operational during peak business hours is the primary objective of every business. The enhanced resiliency of a DB2 instance, in the face of certain categories of critical errors and traps, helps to achieve this objective. If a database instance needs to be recycled after the occurrence of a sustained critical error or trap, these short outages can now be scheduled after peak business hours.

Related tasks

″Recovering from sustained traps″ in

Related reference

″db2diag - db2diag logs analysis tool ″ in Command Reference

Administration notification and diagnostic logs occupy specified amount of disk space

In Version 9.7, you have the ability to configure how much combined disk space is occupied by both the administration notification and diagnostic log files by specifying the total size with the new **diagsize** database manager configuration parameter.

With this improvement, these log files will only grow to the extent you specify without uncontrollable growth potentially consuming all the available free disk space.

The value of the new **diagsize** database manager configuration parameter decides what form of log files will be adopted. If the value is 0 (default), a single administration notification log file (*instance_name*.nfy) and diagnostic log file (db2diag.log) will be adopted, with the size of each log file limited only by the available free disk space. This was the growth behavior of these log files in previous releases. However, if the value is not 0, a series of 10 rotating administration notification log files and 10 rotating diagnostic log files will be adopted. This nonzero value also specifies the total size of all rotating administration notification log files and all rotating diagnostic log files combined, thereby limiting their total size growth.

The amount of the total disk space allocated to the rotating log files, as a percentage of the value specified with the **diagsize** configuration parameter, differs according to platform in the following ways:

UNIX and Linux

- 90% to rotating diagnostic log files
- 10% to rotating administration notification log files

Windows

- 100% to rotating diagnostic log files, due to administration notification on the Windows platform using the Event Log service

The instance must be restarted for the new value of the **diagsize** configuration parameter to take effect.

Related concepts

″Administration notification log″ in Data Recovery and High Availability Guide and Reference

″DB2 diagnostic (db2diag) log files″ in

Related reference

″diagsize - Rotating diagnostic and administration notification logs configuration parameter″ in Database Administration Concepts and Configuration Reference

Chapter 7. Performance enhancements

Version 9.7 contains numerous performance enhancements that continue to make the DB2 data server an industrial-strength data server solution that is suitable for any size of organization.

The DB2 optimizer has been improved with the following enhancements:

- Access plan reuse (see "Access plan reuse ensures consistent performance")
- Statement concentrator support (see "Statement concentrator enables access plan sharing" on page 42)
- RUNSTATS sampling improvements for statistical views (see "Statistics collection sampling performance for statistical views has been improved" on page 43)
- The ALTER PACKAGE statement for applying optimization profiles (see "Optimization guidelines for packages can be applied more easily" on page 43)
- Cost model improvements for queries in partitioned database environments (see "Cost model has been improved for queries in partitioned database environments" on page 44)

In addition, faster data access and increased data concurrency is provided by the following enhancements:

- Cursor stability (CS) isolation level with currently committed semantics (see "Cursor stability (CS) isolation level enhancements provide more concurrency" on page 44)
- Scan sharing (see "Scan sharing improves concurrency and performance" on page 45)
- Partitioned indexes on partitioned tables (see "Partitioned indexes on partitioned tables improve performance" on page 46)
- Materialized query table (MQT) matching enhancements (see "Materialized query table (MQT) matching includes more scenarios" on page 48)
- The ability to store some inline LOB files in tables (see "Small LOBs can be stored in table rows and compressed" on page 48)
- I/O completion ports (IOCP) support on the AIX operating system (see "I/O completion ports (IOCP) is used by default for asynchronous I/O (AIX)" on page 49)
- Isolation level locking intent support in subselect and fullselect clauses (see "Isolation level locking intent can be specified in subselect and fullselect clauses" on page 50)

Another enhancement extends the support of optimization guidelines to XML data. For more information, see "Optimization profiles support guidelines for XML data" on page 24.

Access plan reuse ensures consistent performance

Starting with Version 9.7, you can now have the query compiler attempt to reuse access plans for static SQL queries.

Access plan reuse causes the access plan chosen for a static SQL statement to stay the same as, or very similar to, the existing query execution plan across binds or rebinds. Users that enable access plan reuse can ensure that their queries will execute with predictable performance across version or fix pack upgrades, updates to statistics, and some configuration parameter changes.

You can enable access plan reuse using the new ALTER PACKAGE statement, or the new APREUSE option for the BIND, REBIND, and PRECOMPILE commands.

Related concepts

″Access plan reuse″ in Troubleshooting and Tuning Database Performance

Related tasks

″Customizing precompile and bind options for compiled SQL objects″ in SQL Procedural Languages: Application Enablement and Support

Related reference

″SYSCAT.PACKAGES ″ in SQL Reference, Volume 1

″BIND ″ in Command Reference

″PRECOMPILE ″ in Command Reference

″REBIND ″ in Command Reference

″Query compiler variables″ in Database Administration Concepts and Configuration Reference

″ALTER PACKAGE ″ in SQL Reference, Volume 2

Statement concentrator enables access plan sharing

The statement concentrator enables SQL statements that are identical except for the values of literals to share the same access plan.

This feature is intended for OLTP workloads where simple statements are repeatedly generated with different literal values. In these workloads, the cost of recompiling the statements many times adds significant overhead. The statement concentrator avoids the compilation overhead by enabling the compiled statement to be reused, regardless of the values of the literals.

The statement concentrator is enabled using the **stmt_conc** database manager configuration parameter.

Java and CLI client applications can enable support for this feature.

Enabling statement concentrator at the client level is preferable to the database manager level for several reasons. Firstly, it allows statement concentrator to be controlled at the finest level. Secondly, it is the only consistent way to enable statement concentrator throughout the DB2 family of products.

Related concepts

“JDBC and SQLJ support has been enhanced” on page 92

″Statement concentrator reduces compilation overhead″ in Troubleshooting and Tuning Database Performance

Related reference

″Statement attributes (CLI) list″ in Call Level Interface Guide and Reference, Volume 2

″Connection attributes (CLI) list″ in Call Level Interface Guide and Reference, Volume 2

″DB2Connection interface″ in Developing Java Applications

″IBM Data Server Driver for JDBC and SQLJ properties for DB2 Database for Linux , UNIX , and Windows″ in Developing Java Applications

″stmt_conc - Statement concentrator configuration parameter″ in Database Administration Concepts and Configuration Reference

″StmtConcentrator CLI/ODBC configuration keyword″ in Call Level Interface Guide and Reference, Volume 2

Statistics collection sampling performance for statistical views has been improved

Starting in Version 9.7, the performance of sampling RUNSTATS on statistical views has been enhanced. The SYSTEM sampling style is now supported and the overall sampling performance has been improved.

In previous releases, you could only use the BERNOULLI sampling, which uses row-level sampling to collect view statistics. Now you can use system sampling, which uses page-level sampling to collect view statistics, where it is permitted. Page-level sampling provides excellent performance because only a sample of the pages is read.

In addition, the sampling performance has been improved for statistical views whose definitions are a select over a single base table or referential integrity joins among foreign key tables and primary key tables. The performance benefit occurs even if the referential integrity constraint is informational. The referential integrity constraint allows the sampling specification to be applied directly to the foreign key tables rather than to the statistical view result.

Related concepts

″Data sampling in queries″ in Troubleshooting and Tuning Database Performance

Related reference

″RUNSTATS ″ in Command Reference

Optimization guidelines for packages can be applied more easily

In Version 9.7, you can use the new ALTER PACKAGE statement to associate an optimization profile with a package without performing a BIND operation. This statement enables you to apply optimization profiles more easily.

For dynamic SQL statements, guidelines within the optimization profile are applied right away.

For static SQL statements, the guidelines are applied at the next explicit or implicit REBIND operation. Applying guidelines to these packages is simplified as you no longer need to find the BIND file or remember the BIND options. You can use the ALTER PACKAGE statement and then issue the REBIND command.

Related reference

″ALTER PACKAGE ″ in SQL Reference, Volume 2

Cost model has been improved for queries in partitioned database environments

The optimizer's cost model has been enhanced to improve performance of some queries in partitioned database environments.

The updates to the optimizer's cost model for processing in partitioned database environments now better reflect the CPU cost associated with transferring data between database partitions. This improvement will have a positive impact on the performance of data partitioned queries where the CPU costs of data transmission contribute significantly to query execution time.

Cursor stability (CS) isolation level enhancements provide more concurrency

In Version 9.7, you can use the CS isolation level with currently committed semantics to significantly reduce lock wait and deadlock scenarios. It is the default for new databases.

In previous versions, CS prevented an application from reading any row that was changed by other applications until the change was committed. In Version 9.7, under CS, a read operation does not necessarily wait for a change to a row to be committed before returning a value. Where possible, a read operation now returns the currently committed result, ignoring what might happen to an uncommitted operation. An example of an exception is for updatable cursors; in that case, currently committed results cannot be returned immediately if the row might be updated based on its previous contents.

The new CS behavior is beneficial in high-throughput transaction-processing database environments. In such environments, waiting on locks cannot be tolerated. This new behavior is particularly beneficial if your applications run against databases from multiple vendors. You can use CS instead of writing and maintaining code pertaining to locking semantics specifically for DB2 databases.

This new CS behavior is disabled for existing databases that you upgrade from a previous release. You can enable or disable the behavior by using the new database configuration parameter **cur_commit**. Also, you can override the database-level setting for individual applications using the **CONCURRENTACCESSRESOLUTION** option of the BIND and PRECOMPILE commands. You can override the database-level setting for stored procedures using the **DB2_SQLROUTINE_PREPOPTS** registry variable and the SET_ROUTINE_OPTS procedure.

Currently committed semantics apply only to read-only scans that do not involve catalog tables or the internal scans that are used to evaluate constraints. Note that, because currently committed is decided at the scan level, a writer's access plan might include currently committed scans. For example, the scan for a read-only subquery can involve currently committed semantics. Because currently committed semantics obey isolation level semantics, applications running under currently committed semantics continue to respect isolation levels.

Currently committed semantics require more log space because additional space is required for logging the first update of a data row during a transaction. This log data is required for retrieving the currently committed image of the row. Depending on the workload, the additional log data can have an insignificant or measurable impact on the total log space used. The requirement for additional log space does not apply when **cur_commit** is disabled.

You can use the AIRLINE.war sample program to learn how to use this feature.

Related concepts

″Isolation levels″ in Troubleshooting and Tuning Database Performance

"New DB2 sample programs have been added" on page 89

″Currently committed semantics improve concurrency″ in Troubleshooting and Tuning Database Performance

Related reference

″BIND ″ in Command Reference

″PRECOMPILE ″ in Command Reference

″Query compiler variables″ in Database Administration Concepts and Configuration Reference

″cur_commit - Currently committed configuration parameter″ in Database Administration Concepts and Configuration Reference

Scan sharing improves concurrency and performance

Starting with DB2 Version 9.7, *scan sharing* provides the ability of multiple scanners to coordinate the shared use of buffer pool pages and potentially reduce I/O through buffer pool page retention. Scan sharing increases workload concurrency and performance without requiring any expensive hardware upgrades.

Scan sharing provides the following benefits:

- The system can support a larger number of concurrent applications.
- Queries can perform better.
- System throughput can increase, benefiting even queries that do not participate in scan sharing.

Scan sharing is useful in situations where the system might not be optimally tuned (for example, tuning experts are not available, there is insufficient time to tune, or you cannot tune the system for specific queries) or the system might be I/O bound (for example, you might have many queries performing data scans or an old I/O system). Scan sharing is particularly effective in environments with applications that perform scans such as table scans or MDC block index scans of large tables.

The compiler determines whether a scan is eligible to participate in scan sharing based on criteria such as the type of scan, its purpose, the isolation level, and the amount of work that is done per record. The data server manages shared scans in *share groups* and tries to keep scans in the same group together as long as possible to maximize the benefits obtained by sharing buffer pool pages. Sometimes, however, the data server regroups scans to optimize scan sharing.

The data server measures the *distance* between two scans in the same share group based on the number of buffer pool pages that lies between them. If the distance between two scans in the same share group grows too large, they might not be able to share buffer pool pages.

The data server also monitors the speed of the scans. For example, assume that one scan is faster than another. In this situation, buffer pool pages that are accessed by the first scan might be cleared from the buffer pool before another scan in the share group can access them. To avoid this situation, the data server might throttle the faster scan, which allows slower scans to access the data pages before they are cleared. A high-priority scan, however, is never throttled by a lower priority scan and might move to another share group instead. By either throttling the fast scan or by moving it to a faster share group, the data server adjusts the share groups to ensure that sharing remains optimized.

The method by which buffer pool pages are returned to the pool of available pages also helps to optimize scan sharing. Pages released by the trailing scan of a group are returned to the pool before pages released by the leading scan of a group. Pages from the trailing scan can be released first because the share group has no scans after the trailing scan that will read the pages; however, other scans might have to access the pages used by leading scan, so these pages are released later.

In some situations, a scan might access the buffer pages of another scan part way through the list of pages, then return to the top of the list after reaching the last page. This type of scan is known as a *wrapping scan*.

You can use the db2pd command to view information about scan sharing. For example, for an individual shared scan, the db2pd output shows data such as the scan speed and the amount of time that the scan was throttled. For a sharing group, the command output shows the number of scans in the group and the number of pages shared by the group.

The EXPLAIN_ARGUMENT table has new rows to contain scan-sharing information about table scans and index scans. You can use the db2exfmt command to format and view the contents of this table.

You can use optimizer profiles to override decisions that the compiler makes about scan sharing. However, it is recommended that you not use these optimizer profiles unless recommended to do so by DB2 Service.

Related concepts

″Access types″ in Troubleshooting and Tuning Database Performance

″Scan sharing″ in Troubleshooting and Tuning Database Performance

Related reference

″EXPLAIN_ARGUMENT table″ in SQL Reference, Volume 1

″db2exfmt - Explain table format ″ in Command Reference

″db2pd - Monitor and troubleshoot DB2 database ″ in Command Reference

Partitioned indexes on partitioned tables improve performance

In Version 9.7, you can have indexes that refer to rows of data across all partitions in a partitioned table (known as *nonpartitioned* indexes), or you can have the index itself partitioned such that each data partition has an associated *index partition*. You can also have both nonpartitioned and partitioned indexes for partitioned tables.

An index on an individual data partition is an index partition; the set of index partitions that make up the entire index for the table is a *partitioned index*.

Before Version 9.7, if you used an ALTER TABLE statement to attach a source table to a partitioned table as a new partition, the data in the new partition was not visible until after you issued a SET INTEGRITY statement to perform tasks such as updating indexes, enforcing constraints, and checking ranges. If the source table that you attached had a large amount of data, SET INTEGRITY processing might be slow and use a considerable amount of log space. Access to the data might be delayed.

Starting in Version 9.7, you can use partitioned indexes to improve performance when you roll data into a table. Before you alter a partitioned table that uses partitioned indexes to attach a new partition or a new source table, you should create indexes on the table that you are attaching to match the partitioned indexes of the partitioned table. After attaching the source table, you still must issue a SET INTEGRITY statement to perform tasks such as range validation and constraint checking. However, if the source tables indexes match all of the partitioned indexes on the target table, SET INTEGRITY processing does not incur the performance and logging overhead associated with index maintenance. The newly rolled-in data is accessible quicker than it would otherwise be.

Partitioned indexes can also improve performance when you roll data out of a table. When you alter the table to detach one of its data partitions, that data partition takes its partitioned indexes with it, becoming a stand-alone table with its own indexes. You do not have to recreate the indexes for the table after detaching the data partition. Unlike nonpartitioned indexes, when you detach a data partition from a table that uses partitioned indexes, the associated index partitions go with it. As a result, there is no need for asynchronous index cleanup (AIC).

In addition, partition elimination for queries against a partitioned table that uses partitioned indexes can be more efficient. For nonpartitioned indexes, partition elimination can only eliminate data partitions. For partitioned indexes, partition elimination can eliminate both data and index partitions. This can result in having to scan fewer keys and index pages than a similar query over a nonpartitioned index.

By default, when you create indexes on partitioned tables, they are partitioned indexes. You can also include the PARTITIONED keyword of the CREATE INDEX statement to have a partitioned index created. You must use the NOT PARTITIONED keywords if you want a nonpartitioned index. All partitioned indexes for a data partition are stored in the same index object, regardless of whether the index partitions are stored in the same table space used for the data partition or in a different table space.

As in previous releases, you can use the ALTER TABLE statement with the ADD PARTITION clause to create a new data partition for a partitioned table. To specify that partitioned indexes on the new data partition are to be stored in a different table space than the table space used for the data partition, use the INDEX IN option of the ADD PARTITION clause. If partitioned indexes exist on the partitioned table, the ADD PARTITION operation extends these indexes to the new partition, and the partitioned indexes are stored in the table space that you specify. If you do not use the INDEX IN option, the partitioned indexes are stored in the same table space in which the new data partition is stored.

Partitioned indexes are not supported for user-created indexes over XML data.

Related concepts

″Partitioned tables″ in Partitioning and Clustering Guide

″Optimization strategies for partitioned tables″ in Troubleshooting and Tuning Database Performance

Related reference

″ALTER TABLE ″ in SQL Reference, Volume 2

″CREATE INDEX ″ in SQL Reference, Volume 2

Materialized query table (MQT) matching includes more scenarios

Version 9.7 includes enhancements that improve MQT matching capabilities which will improve query performance.

In previous versions, the optimizer would choose an MQT based on its cost model. You can now override the optimizer decision by forcing it to choose specific MQTs with the new MQTENFORCE element using optimization profiles.

In addition, MQTs will be matched in the following new scenarios:

- An MQT that specifies a view, possibly containing a complex query, can be matched to a query that uses that view. In previous releases, queries that specified a view with a construct such as OUTER JOIN or some complex UNION ALL query could not be matched to an MQT. You can now create views for the portion of queries which is not matchable and then create MQTs that do a simple SELECT operation from these views. If the queries also reference these views, the optimizer will consider replacing the view with the corresponding MQT.
- Queries containing a SELECT DISTINCT or GROUP BY clause can be matched to MQTs whose definitions start with SELECT DISTINCT.
- Queries containing an EXISTS predicate can also be matched to MQTs with an appropriate EXISTS predicate.
- Some additional scenarios involving datetime functions are better matched. For example, queries containing a QUARTER function can be matched to an MQT returning MONTH. Another example is when a query contains DAYOFWEEK function and the MQT contains DAYOFWEEK_ISO (or the reverse scenario).
- Other previously unmatched scenarios are now matched when referential integrity constraints (or informational referential integrity) are defined.

Small LOBs can be stored in table rows and compressed

When a LOB is smaller than a specified size, the LOB is now stored in the row of the base table instead of in the separate LOB storage object. Oversize LOBs are stored transparently in the LOB storage object. This support is already available for small XML documents.

If you work mostly with small LOBs, base table row storage provides increased performance for any operation that queries, inserts, updates, or deletes LOBs, because fewer I/O operations are required. If you also use data row compression, LOBs are compressed, which reduces storage space requirements and improves I/O efficiency for LOBs.

The maximum size of LOBs to store in the base table is specified by the INLINE LENGTH option of the CREATE TABLE statement and the ALTER TABLE statement. You can specify a value up to 32 673 bytes (the page size used affects this value).

Row storage of LOBs is similar to how a structured type instance or an XML document can be stored inline in the row of a table.

Version 9.7 includes two functions that provide information about the base table storage of XML documents and LOB data and help you manage them:

ADMIN_IS_INLINED
After you have enabled base table row storage, you can use ADMIN_IS_INLINED to determine if XML documents or LOB data are being stored in the base table row.

ADMIN_EST_INLINE_LENGTH
You can use ADMIN_EST_INLINE_LENGTH to display the size of the XML or LOB data and use the information when enabling base table row storage or adjusting the size used for base table row storage.

You can use a new sample program to learn how to use the table functions with XML documents.

Related concepts

“New DB2 sample programs have been added” on page 89

″Inline LOBs improve performance″ in Troubleshooting and Tuning Database Performance

″Storing LOBs inline in table rows″ in Database Administration Concepts and Configuration Reference

Related reference

″ALTER TABLE ″ in SQL Reference, Volume 2

″CREATE TABLE ″ in SQL Reference, Volume 2

″Large objects (LOBs)″ in SQL Reference, Volume 1

I/O completion ports (IOCP) is used by default for asynchronous I/O (AIX)

When I/O completion ports are configured, they are used by default to process asynchronous I/O (AIO) requests on the AIX operating system and can improve overall system performance.

AIX operating systems in symmetric multi-processor (SMP) environments with more than 16 processors can use IOCP for their AIO processing. The AIO server processes from the AIX operating system manage the I/O requests by processing large numbers of requests in the most optimal way for the system.

Your operating system might not have the IOCP module installed and configured. If the DB2 Version 9.7 minimum AIX operating system requirements were applied using an operating system upgrade rather than using a new operating system installation, you must install and configure the IOCP module separately.

Related tasks

"Configuring IOCP on AIX" in Troubleshooting and Tuning Database Performance

Isolation level locking intent can be specified in subselect and fullselect clauses

Starting in Version 9.7, you can now specify different isolation levels and locking intents for tables in different subselect clauses.

In previous releases, you could only specify the isolation level and locking intent at the statement level, meaning that those settings would be applied to all the tables in the SQL statement. This change allows for improved concurrency because you can specifically control isolation levels instead of using a high isolation level for the entire query.

Example

In the following example of an inner join between two tables, the repeatable read (RR) isolation level specified for table MySchema.T1 will override the read stability (RS) isolation level specified for the session:

```
SET CURRENT ISOLATION LEVEL RS;
SELECT T1.c1, T1.c2, T2.c3
  FROM (SELECT c1, c2 FROM MySchema.T1 WITH RR) as T1,
            MySchema.T2 as T2
  WHERE T1.c1 = T2.c4;
```

MySchema.T1 will be accessed using repeatable read and MySchema.T2 will be accessed using isolation read stability.

Chapter 8. SQL compatibility enhancements

If you work with Oracle databases, you will find that Version 9.7 contains many new enhancements that make the DB2 products more familiar to you. These enhancements also make it particularly easy to enable Oracle applications to run quickly in DB2 environments.

For example, Version 9.7 introduces the following application and administration interfaces that will be familiar to Oracle users:

- CLPPlus, an interactive command-line processor that is compatible with Oracle SQL*Plus (see “SQL*Plus compatible command line processor has been added”)
- DB2 catalog information that presents information in a way that is compatible with how Oracle data dictionary views present information (see “Oracle data dictionary-compatible view support has been added”)
- System-defined modules, which are compatible with Oracle built-in packages (see “System-defined modules simplify SQL PL and application logic” on page 86)

In addition, if you also work with Oracle software, you will be familiar with several new SQL constructions and semantics when writing and running Version 9.7 applications:

- PL/SQL code compilation and execution in Version 9.7 environments (see “PL/SQL language is supported by DB2 interfaces” on page 52)
- Data type support for NUMBER, VARCHAR2, and DATE with the time component (see “Data type support has been extended” on page 53)
- The ROWNUM and ROWID pseudocolumns (see “Alternative SQL language syntax is supported” on page 53)

You control which of these compatibility features are enabled using specific settings of the **DB2_COMPATIBILITY_VECTOR** registry variable. For more information, see “Oracle-compatible mode can be enabled” on page 54.

SQL*Plus compatible command line processor has been added

The CLPPlus processor is a new, easy to use, interactive command line processor for SQL statements and database commands. The processor provides support for dynamically creating, editing, and running SQL statements and scripts.

A set of commands is supported for connecting to databases, manipulating a statement or script defined in a buffer, modifying lines in the buffer, formatting result sets, and performing other related administrative tasks. This processor provides compatible features to the SQL*Plus command line processor.

Oracle data dictionary-compatible view support has been added

Views that present DB2 catalog information in a way that is compatible to Oracle data dictionary views are now available. The data dictionary is a repository for database metadata.

There are three different versions of each data dictionary view, and each version is identified by the prefix of the view name. Not all versions apply to each view.

- USER_* views return information about objects that are owned by the current database user.
- ALL_* views return information about objects to which the current user has access.
- DBA_* views return information about all objects in the database, regardless of who owns them.

The SELECT privilege is granted to PUBLIC for all USER_* and ALL_* views.

The compatible views are based on DB2 system catalog views and snapshot monitor SQL administrative views. Appropriate privileges on the underlying catalog and snapshot monitor views are required to use the Oracle data dictionary-compatible views.

The data dictionary views are self-describing. The DICTIONARY view returns a listing of all data dictionary views with comments that describe the content of each view. The DICT_COLUMNS view returns a list of all columns in all data dictionary views. With these two views, you can determine what information is available and how to access it.

Oracle data dictionary-compatible view support is enabled through the setting of the **DB2_COMPATIBILITY_VECTOR** registry variable.

For the complete list of the supported views, see the “Oracle data dictionary-compatible views” topic.

Related concepts

″Oracle data dictionary-compatible views″ in SQL Procedural Languages: Application Enablement and Support

Related reference

″DB2_COMPATIBILITY_VECTOR registry variable″ in SQL Procedural Languages: Application Enablement and Support

PL/SQL language is supported by DB2 interfaces

In DB2 Version 9.7, PL/SQL can be compiled and executed using DB2 interfaces.

DB2 Version 9.7 supports the compilation of the PL/SQL language. This support for PL/SQL compilation can be enabled by setting or updating the **DB2_COMPATIBILITY_VECTOR** registry variable.

Porting existing PL/SQL scripts, package definitions, or individual PL/SQL and SQL statements that work with other database management systems to DB2 Version 9.7 is done by executing them from a DB2 Command Window or from the DB2 Command Line Processor.

With the DB2 environment initialized for PL/SQL, the DB2 compiler processes the PL/SQL DDL statements and creates the required database objects in the database. PL/SQL anonymous blocks, as well as references to commonly used package routines and views can also be compiled and executed.

With this support, you quickly enable PL/SQL solutions in DB2 environments.

Related concepts

″PL/SQL support″ in SQL Procedural Languages: Application Enablement and Support

Related reference

″DB2_COMPATIBILITY_VECTOR registry variable″ in SQL Procedural Languages: Application Enablement and Support

Data type support has been extended

To make data management more compatible with the Oracle database management system, you can now use the NUMBER and VARCHAR2 data types. You can also have the database manager interpret the DATE data type (normally comprised of year, month, day) as a TIMESTAMP(0) data type (composed of year, month, day, hour, minute, second).

These data types are enabled by setting the **DB2_COMPATIBILITY_VECTOR** registry variable.

This support includes Oracle-compatible functions for casting data types and performing data type arithmetic on the DATE data type.

You can use the datecompat.db2 sample program to learn how to use some of these features.

Related concepts

"New DB2 sample programs have been added" on page 89

Related reference

″DB2_COMPATIBILITY_VECTOR registry variable″ in SQL Procedural Languages: Application Enablement and Support

″NUMBER data type″ in SQL Procedural Languages: Application Enablement and Support

″VARCHAR2 data type″ in SQL Procedural Languages: Application Enablement and Support

″DATE data type based on TIMESTAMP(0)″ in SQL Procedural Languages: Application Enablement and Support

Alternative SQL language syntax is supported

Starting in Version 9.7, you can use an alternative syntax for the DELETE statement and the RID_BIT scalar function to accommodate similar features supported by other database vendors.

Version 9.7 now supports a simplified version of the DELETE FROM statement, such that the FROM clause can be omitted. Although the FROM clause in DELETE is optional, omitting FROM is nonstandard and is not recommended.

The RID_BIT scalar function also supports an alternative syntax using a pseudocolumn, ROWID. An unqualified reference to ROWID is equivalent to `RID_BIT()`, and a qualified reference to ROWID, such as EMPLOYEE.ROWID, is equivalent to `RID_BIT(EMPLOYEE)`.

Other alternative SQL syntax, such as using ROWNUM as a synonym for ROW_NUMBER() OVER(), can be selectively enabled by setting the **DB2_COMPATIBILITY_VECTOR** registry variable.

Related concepts

″Introduction to DB2 compatibility features″ in SQL Procedural Languages: Application Enablement and Support

Related reference

″DELETE ″ in SQL Reference, Volume 2

″RID_BIT and RID ″ in SQL Reference, Volume 1

″ROWNUM pseudocolumn″ in SQL Procedural Languages: Application Enablement and Support

Oracle-compatible mode can be enabled

To make the DB2 environment more compatible with the Oracle database management system environment, you can set the new **DB2_COMPATIBILITY_VECTOR** registry variable.

This registry variable is represented as a hexadecimal value, and each bit in the variable enables one of the DB2 compatibility features. Depending on the application that you want to enable for the DB2 data server, you can turn on compatibility for data types, semantics, and behaviors that the application might require.

Related tasks

″Setting up DB2 for Oracle application enablement″ in SQL Procedural Languages: Application Enablement and Support

Related reference

″DB2_COMPATIBILITY_VECTOR registry variable″ in SQL Procedural Languages: Application Enablement and Support

Chapter 9. Workload management enhancements

DB2 Version 9.7 features extend the workload management capabilities provided in previous releases.

Workload activity-level control, the definition of workloads, and aggregate activity data collection have been both enhanced and simplified in Version 9.7 by the following functionality:

- Activity-based thresholds (see "Workloads support activity-based threshold controls")
- Wild card and IP addresses support (see "Workload connection attributes have been enhanced" on page 56)
- Workload-level aggregate activity data collection and new high watermarks (see "Workload-level aggregate activity data collection and new high watermarks provide additional statistics" on page 57)

Controlling the relative priority for work with service classes has been enhanced by the following Version 9.7 functionality:

- Buffer pool I/O priority control for service classes (see "Buffer pool I/O priority can be controlled for service classes" on page 58)
- Integration with Linux workload management (see "DB2 workload manager supports integration with Linux workload management (WLM)" on page 58)

To help maintain stability on your data server, you can use the following threshold enhancements to control additional resources and exert more precise control over existing ones:

- New thresholds for controlling specific resources (see "New thresholds provide additional activity control" on page 58)
- 1-minute check interval support for a subset of time-based thresholds (see "Time-based thresholds support 1-minute check intervals" on page 59)

You can now configure DB2 workload manager to automatically lower the priority of in-progress activities over time. For more information, see "Support of priority aging of in-progress activities has been added" on page 59.

Additional features added in Version 9.7 improve workload monitoring. For more information, see Chapter 5, "Monitoring enhancements," on page 29.

Workloads support activity-based threshold controls

Support for defining activity-based thresholds on the workload domain has been added to some existing thresholds and included with some new thresholds. When you define workloads, you can apply these activity-based thresholds for greater control over resources.

Using activity-based threshold controls on the workload domain also means that you no longer have to isolate applications from each other in separate service classes to apply a specific set of thresholds to a given application, which simplifies your DB2 Workload Manager configuration.

You can define the following thresholds on the workload domain:

ESTIMATEDSQLCOST
Specifies the maximum estimated cost for DML activities

SQLROWSRETURNED
Specifies the maximum number of rows that the data server can return to the client

ACTIVITYTOTALTIME
Specifies the maximum lifetime of an activity

SQLTEMPSPACE
Specifies the maximum amount of system temporary table space that a DML activity can use on a particular database partition

SQLROWSREAD
Specifies the maximum number of rows that a DML activity can read on any database partition

CPUTIME
Specifies the maximum amount of combined user and system processor time that an activity can use on a particular database partition while the activity is running.

Related concepts

″Example: Using thresholds″ in Workload Manager Guide and Reference

Workload connection attributes have been enhanced

You can now specify an asterisk (*) as a wild card for some workload connection attributes, and specify IPv4 addresses, IPv6 addresses, and secure domain names in the ADDRESS connection attribute when you create or alter a workload.

Workload connection attributes support wild cards

Using wild cards makes it easier to define workloads. If you know that there is incoming work with similar connection attribute values that can all be assigned to the same workload, use a wild card as part of the workload connection attribute value. For example, if you have a number of accounts receivable applications that share a similar name (accrec01, accrec02 ... accrec15) and that should be assigned to the same workload, you can define a workload with an application name connection attribute of accrec* which will match all these applications names instead of defining a connection attribute for each application name.

The following connection attributes support the use of wild cards:

APPLNAME
Application name

CURRENT CLIENT_ACCTNG
Client accounting string

CURRENT CLIENT_APPLNAME
Client application name

CURRENT CLIENT_USERID
Client user ID

CURRENT CLIENT_WRKSTNNAME
Client workstation name

The Workload ADDRESS connection attribute supports IP addresses

Workloads identify incoming work based on the attributes of the database connection under which the work is submitted. By being able to specify IP addresses in the ADDRESS connection attribute, you have an additional means for determining the source of incoming work at your disposal, which you can use to assign work to the correct workload.

Related concepts

″Work identification by origin with workloads″ in Workload Manager Guide and Reference

Workload-level aggregate activity data collection and new high watermarks provide additional statistics

To support priority aging of in-progress activities and the definition of thresholds at the workload level, enhanced monitoring and statistics are available.

Collection of aggregate activity data at the workload level

Aggregate activity data can now be accumulated at the workload level in addition to the service class and work class levels. This offers the following benefits:

- Increased granularity for monitoring. For example, you can now acquire histogram information for workload definitions and for specific applications in addition to acquiring similar information at the service class or work class level.
- Simplified view of aggregate data when ongoing work is priority aged. Priority aging, with which you can lower the priority of work over time by remapping the work between service subclasses, can make the collection of meaningful aggregate activity statistics across service subclasses complex. Through the support of aggregate activity statistics at the workload level, the collection of these statistics is simplified.
- Easier determination of maximum values for workload thresholds. You can use the aggregate activity data collected at the workload level to help determine appropriate maximum values for thresholds defined on the workload domain.

New high watermarks

To make it easier for you to determine what threshold values you should use with the new CPUTIME and SQLROWSREAD thresholds, two new high watermarks are available:

act_cpu_time_top
The high watermark for processor time used by activities at all nesting levels in a service class, workload, or work class

act_rows_read_top
The high watermark for the number of rows read by activities at all nesting levels in a service class, workload, or work class

You can also determine the highest lock wait time of any request on a partition for a workload during a time interval by using the following new high watermark:

lock_wait_time_top
The high watermark for lock wait times for any request in a workload, in milliseconds

Related concepts

″Statistics for DB2 workload manager objects″ in Workload Manager Guide and Reference

"Workloads support activity-based threshold controls" on page 55

Chapter 5, "Monitoring enhancements," on page 29

Buffer pool I/O priority can be controlled for service classes

You can use buffer pool I/O priority settings to influence the proportion of pages in the buffer pool that may be occupied by activities in a given service class, which can improve the throughput and performance of activities in that service class.

You can use the buffer pool I/O priority resource control with the existing agent priority and prefetch priority controls for any user-defined service class to control the relative priority of each service class.

Related concepts

″Resource assignment with service classes″ in Workload Manager Guide and Reference

″Buffer pool priority of service classes″ in Workload Manager Guide and Reference

DB2 workload manager supports integration with Linux workload management (WLM)

You can use the optional integration between DB2 service classes and Linux WLM classes (with Linux kernel version 2.6.26 or later) to control the amount of processor resource allocated to each service class. When you use this integration, you take advantage of the native control capabilities provided by Linux WLM.

The additional functionality provided by integrating DB2 workload manager with Linux WLM is similar to what is provided when you integrate DB2 workload manager with AIX Workload Manager. If enabled,all threads running in a DB2 service class are mapped to a Linux WLM class, where they are subject to the processor resource controls that you define in Linux WLM.

Related concepts

″Integration of Linux workload management with DB2 workload manager″ in Workload Manager Guide and Reference

″Integration with operating system workload managers″ in Workload Manager Guide and Reference

New thresholds provide additional activity control

You can exercise additional control over activities on your data server with new thresholds.

You can use the following new thresholds to help maintain stability on your data server by controlling a specific resource:

AGGSQLTEMPSPACE
Controls the maximum amount of system temporary table space that can be used across all activities in a service subclass

CPUTIME
Controls the maximum amount of processor time that an activity can use on a particular database partition during execution

SQLROWSREAD
Controls the maximum number of rows that an activity can read on a particular database partition

Related concepts

″Priority aging of ongoing work″ in Workload Manager Guide and Reference

″Example: Using thresholds″ in Workload Manager Guide and Reference

“Support of priority aging of in-progress activities has been added”

Time-based thresholds support 1-minute check intervals

The granularity for time-based thresholds has been improved by lowering the minimum available time interval from 5 minutes to 1 minute. This change helps reduce delays when detecting excessive time consumption early is important.

The following time-based thresholds use the improved granularity:

ACTIVITYTOTALTIME
Controls the maximum lifetime of an activity

CONNECTIONIDLETIME
Controls the maximum amount of time that a connection can be idle, that is, not working on a user request

Related concepts

″Example: Using thresholds″ in Workload Manager Guide and Reference

Related reference

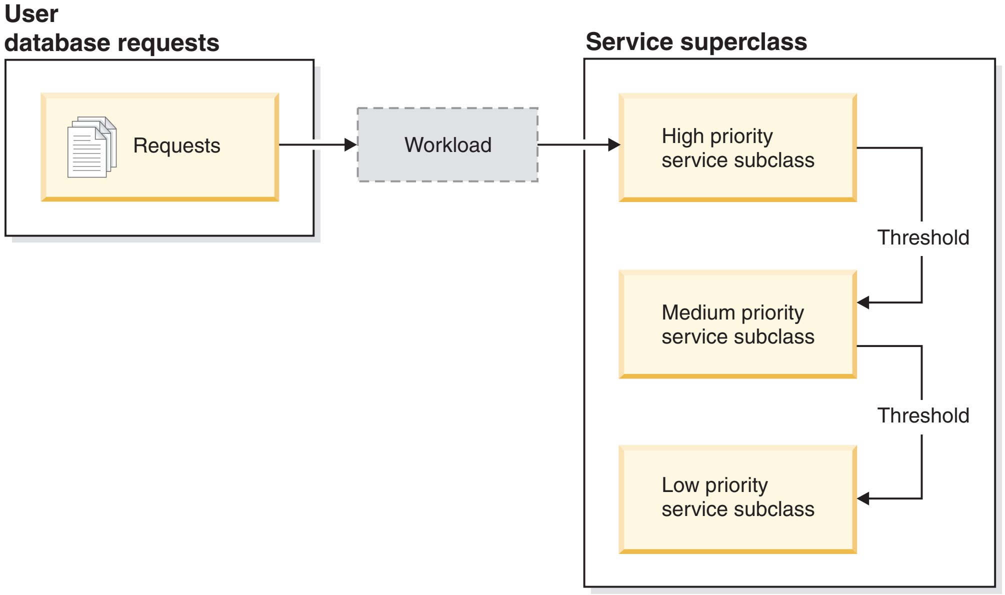
″CREATE THRESHOLD ″ in Workload Manager Guide and Reference

Support of priority aging of in-progress activities has been added

You can now configure DB2 workload manager to automatically lower the priority of in-progress activities over time, known as priority aging. You use priority aging to decrease the priority of longer-running activities, so that throughput for shorter-running activities can be improved.

Your data server changes the priority of in-progress activities by moving them (remapping them) between service subclasses in response to the amount of processor time used or the number of rows read. You can automatically decrease, or age, the priority of longer-running activities by having them remapped to a service class with fewer resources (that is, with more resource constraints). Because the data server performs the remapping automatically, you do not require detailed knowledge of the activities running on your data server at a particular time.

The following illustration shows how activities are priority aged:



As work enters the system, activities are automatically placed into the first service class and begin running using the high-priority settings of this service class. If you also define thresholds for each of the service classes that limit the time or resources used during execution, activities are dynamically reassigned to the next-lower service class if the threshold of the next-higher class is violated. This dynamic resource control is repeatedly applied until activities complete or are in the lowest-priority class, where they remain until it is completed or you force them to stop running.

New thresholds and new threshold action

Two new thresholds support priority aging by providing a remapping action. Activities can be remapped when a threshold that has a REMAP ACTIVITY action defined for it is violated.

CPUTIMEINSC
Controls the maximum amount of processor time that an activity can use in a particular service subclass on a particular database partition

SQLROWSREADINSC
Controls the maximum number of rows that an activity can read on a particular database partition while the activity is running in a particular service subclass

Enhanced monitoring and statistics collection

To support priority aging, enhanced monitoring and statistics collection are available.

You can use three new monitor elements to identify remapped activities and affected service subclasses:

num_remaps
Indicates how many times an activity has been remapped

act_remapped_in
Counts the number of activities to be remapped into a service subclass

act_remapped_out

Counts the number of activities to be remapped out of a particular service subclass

The collection of aggregate activity statistics when you remap activities from one service subclass to another is kept simple by the introduction of aggregate activity statistics collection at the workload level. You can use this feature to gather detailed statistics about activities without having to aggregate statistics across service subclasses that are involved in remapping.

Sample priority aging scripts, sample scenarios

You can use two sample scripts included with your data server installation to exploit priority aging of ongoing work quickly. These scripts create all the required workload management objects for moving ongoing work between tiered service classes for you. You can use another script to drop all the objects created from the sample scripts.

Common scenarios for how you might use the scripts on your data server are also provided.

Related concepts

″Priority aging of ongoing work″ in Workload Manager Guide and Reference

"New DB2 sample programs have been added" on page 89

″Control of work with thresholds″ in Workload Manager Guide and Reference

Chapter 10. Security enhancements

With the number of both internal and external security threats growing, it is important to separate the tasks of keeping data secure from the management tasks of administering critical systems. Building on the enhancements introduced in previous versions, Version 9.7 enhancements ensure that your sensitive data is even better protected.

Advances in separation of duties provide the following functionality:

- Database administrators (DBAs) maintain full control over database operations, without any access to the data.
- You can easily isolate all security and auditing control into the hands of a separate security administrator.
- You can use new authorities, making application development privileges simpler as well as workload management.

For more information, see "DB2 authorization model has been enhanced to allow separation of duties."

The following enhancements also extend security functionality:

- AES encryption (see "AES encryption of user ID and password enhances security" on page 66)
- Transport Layer Security (see "SSL client support expanded and configuration simplified" on page 66)
- Additional Secure Sockets Layer (SSL) client support (see "SSL client support expanded and configuration simplified" on page 66)

The following enhancements simplify the configuration of security features:

- New database manager configuration parameters and connection string keywords for SSL servers and SSL client configuration (see "SSL client support expanded and configuration simplified" on page 66)
- Transparent LDAP authentication and group lookup support (see "Transparent LDAP authentication and group lookup is supported (AIX)" on page 68)
- Password-setting enhancements (see "Passwords can be the maximum length supported by the operating system" on page 68)

DB2 authorization model has been enhanced to allow separation of duties

Version 9.7 clearly divides the duties of the database administrator and the security administrator and introduces new authorities that enable you to grant only the access a user needs to do their work. These enhancements also make it easier to meet government compliance requirements.

Version 9.7 introduces new authorities for workload management (WLMADM), SQL tuning (SQLADM) and for using the explain facility with SQL statements (EXPLAIN). These authorities allow you to grant users these responsibilities without having to grant them DBADM authority or actual privileges on the base tables, which would give those users more privileges than they need to do their work. Therefore, by using these new authorities, you can minimize the risk of exposing sensitive data.

Version 9.7 also introduces the new authorities DATAACCESS and ACCESSCTRL. DATAACCESS authority is the authority that allows access to data within a specific database. ACCESSCTRL authority is the authority that allows a user to grant and revoke privileges on objects within a specific database. By default, DATAACCESS and ACCESSCTRL authorities are included when the security administrator grants DBADM authority. But if you do not want your database administrator to have access to data, or to be able to grant privileges and authorities, you can choose to not include these authorities.

Note: The creator of a database is automatically granted DBADM, SECADM, DATAACCESS and ACCESSCTRL authorities within that database. If you do not want this user to have any of these authorities, you must revoke them.

Changes for the system administrator (who holds SYSADM authority)

A user who holds SYSADM authority no longer has implicit DBADM authority, so has limited capabilities compared to those available in Version 9.5.

A user who holds SYSADM authority is no longer able to grant any authorities or privileges, except to grant table space privileges.

For a user holding SYSADM authority to obtain the same capabilities as in Version 9.5 (other than the ability to grant SECADM authority), the security administrator must explicitly grant them DBADM authority. Note that when the security administrator grants DBADM authority, the new DATAACCESS and ACCESSCTRL authorities are included by default. This will give the user equivalent Version 9.5 capability. For this user to also be able to grant SECADM authority, they must be granted SECADM authority as well. Note, however, that holding SECADM authority will allow this user to perform more actions than they could as a Version 9.5 system administrator. For example, they will be able to create objects such as roles, trusted contexts and audit policies.

On Windows systems, when the **sysadm_group** database manager configuration parameter is not specified, the LocalSystem account is considered a system administrator (holding SYSADM authority). Any DB2 application that is run by LocalSystem is affected by the change in scope of SYSADM authority in Version 9.7. These applications are typically written in the form of Windows services and run under the LocalSystem account as the service logon account. If there is a need for these applications to perform database actions that are no longer within the scope of SYSADM, you must grant the LocalSystem account the required database privileges or authorities. For example, if an application requires database administrator capabilities, grant the LocalSystem account DBADM authority using the GRANT (Database Authorities) statement. Note that the authorization ID for the LocalSystem account is SYSTEM.

Changes for the security administrator (who holds SECADM authority)

A user who holds SECADM authority can now grant and revoke all authorities and privileges including DBADM and SECADM authorities.

SECADM authority can now be granted to roles and groups (in Version 9.5, SECADM could be granted only to a user).

SECADM authority is no longer necessary to run the audit stored procedures and table functions:

- AUDIT_ARCHIVE
- AUDIT_LIST_LOGS
- AUDIT_DELIM_EXTRACT

In Version 9.7, EXECUTE privilege is sufficient to run these routines, however, only the security administrator can grant the EXECUTE privilege on these routines. This change allows the security administrator to delegate part of their responsibilities to other users.

Changes for the database administrator (who holds DBADM authority)

The following authorities will continue to be available to the database administrator as long as the user holds DBADM authority, but will be lost if DBADM authority is revoked. Granting DBADM authority no longer additionally grants the following separate database authorities because they are already implicitly vested in the DBADM authority level.

- BINDADD
- CONNECT
- CREATETAB
- CREATE_EXTERNAL_ROUTINE
- CREATE_NOT_FENCED_ROUTINE
- IMPLICIT_SCHEMA
- QUIESCE_CONNECT
- LOAD

When the security administrator grants DBADM authority, they can choose whether to give the database administrator the ability to perform the following operations:

- Accessing data within the database.
- Granting and revoking privileges and authorities.

The security administrator can use the following options of the GRANT DBADM ON DATABASE statement to control these functions:

- WITH ACCESSCTRL
- WITHOUT ACCCESSCTRL
- WITH DATAACCESS
- WITHOUT DATAACCCESS

By default, DATAACCESS and ACCESSCTRL authorities are included if they are not specified.

Related concepts

″Authorization, privileges, and object ownership″ in SQL Reference, Volume 1

″Windows LocalSystem account support″ in Database Security Guide

″Authorities overview″ in Database Security Guide

“System administrator (SYSADM) authority scope has changed” on page 145

“Database administrator (DBADM) authority scope has changed” on page 148

“Security administrator (SECADM) abilities have been extended” on page 147

AES encryption of user ID and password enhances security

In Version 9.7, you can now encrypt the user ID and password using the Advanced Encryption Standard (AES) algorithm with keys 256 bits long.

The user ID and password submitted for authentication to DB2 are encrypted when the authentication method negotiated between the DB2 client and the DB2 server is SERVER_ENCRYPT. The authentication method negotiated depends on the authentication type setting of the **authentication** configuration parameter on the server and the authentication requested by the client. The choice of the encryption algorithm used to encrypt the user ID and password, either DES or AES, depends on the setting of the **alternate_auth_enc** database manager configuration parameter:

- NOT_SPECIFIED (the default) means that the server accepts the encryption algorithm that the client proposes.
- AES_CMP means that if the connecting client proposes DES but supports AES encryption, the server renegotiates for AES encryption. Downlevel clients that do not support AES will still be able to connect using DES.
- AES_ONLY means that the server accepts only AES encryption. If the client does not support AES encryption, the connection is rejected.

Related concepts

″Authentication methods for your server″ in Database Security Guide

Related reference

″authentication - Authentication type ″ in Database Administration Concepts and Configuration Reference

″alternate_auth_enc - Alternate encryption algorithm for incoming connections at server configuration parameter″ in Database Administration Concepts and Configuration Reference

SSL client support expanded and configuration simplified

In DB2 Version 9.7, enhanced support for Secure Sockets Layer (SSL) and its successor, Transport Layer Security (TLS), improves the security of data communication by making it easier to configure your server. Additionally, support is expanded to all non-Java DB2 clients, such as CLI/ODBC, .Net Data Provider, embedded SQL, and CLP.

Note: In this topic, references to SSL also apply to TLS.

TLS version 1.0 (RFC2246) and TLS version 1.1 (RFC4346) are supported.

Configuration enhancements

You no longer need to use separate configuration files to set up SSL support. The parameters that you used to set in the SSLconfig.ini and SSLClientconfig.ini files are now replaced by database manager configuration parameters and connection string keywords.

- There are six new server-side configuration parameters:
  - **ssl_svr_keydb** specifies the fully qualified path of the key database file.
  - **ssl_svr_stash** specifies the fully qualified path of the stash file that holds the encrypted password to the key database.
  - **ssl_svr_label** specifies the label of the digital certificate of the server in the key database.
  - **ssl_svcename** specifies the port that the database server uses to await communications from remote clients using the SSL protocol.
  - **ssl_cipherspecs** (optional) specifies the cipher suites that the server supports.
  - **ssl_versions** (optional) specifies the SSL and TLS versions that the server supports.
- There are two new client-side database manager configuration parameters:
  - **ssl_clnt_keydb** specifies the fully qualified path of the key database file on the client.
  - **ssl_clnt_stash** specifies the fully qualified path of the stash file on the client.
- There are two new connection string keywords for CLI/ODBC applications:
  - **ssl_client_keystoredb** - Set **ssl_client_keystoredb** to the fully-qualified key database file name.
  - **ssl_client_keystash** - Set **ssl_client_keystash** to the fully-qualified stash file name.
- There are three new connection string keywords for DB2 .Net Data Provider applications:
  - **SSLClientKeystoredb** - Set **SSLClientKeystoredb** to the fully-qualified key database file name.
  - **SSLClientKeystash** - Set **SSLClientKeystash** to the fully-qualified stash file name.
  - **security** - Set **security** to SSL.

Setting up SSL connections for CLI/ODBC applications

If you are using the IBM Data Server Driver for ODBC and CLI to connect to a database using SSL, you use the connection string parameters **ssl_client_keystoredb**, and **ssl_client_keystash** to set the path for the client key database and for the stash file, and the connection string parameter **security** to set the protocol to SSL.

If you are using the IBM Data Server Client or IBM Data Server Runtime Client to connect to a database using SSL, you use the connection string parameter **security** to set the protocol to SSL, and you can use either the connection string parameters **ssl_client_keystoredb** and **ssl_client_keystash**, or the client-side database manager configuration parameters **ssl_clnt_keydb** and **ssl_clnt_stash**, to set the path for the client key database and for the stash file.

Setting up SSL connections for .Net Data Provider applications

For .Net Data Provider applications, you use the connection string parameters **SSLClientKeystoredb** and **SSLClientKeystash** to set the path for the client key database and for the stash file, and the connection string parameter **security** to set the protocol to SSL

Setting up SSL connections for CLP clients and embedded SQL clients

The SSL keyword has been added to the CATALOG TCPIP NODE command SECURITY parameter. CLP clients and embedded SQL clients can use this keyword and the client-side database manager configuration parameters **ssl_clnt_keydb**, and **ssl_clnt_stash** to connect to a database using SSL.

Related concepts

″Supported cipher suites″ in Database Security Guide

"SSLconfig.ini and SSLClientconfig.ini files replaced with new database manager configuration parameters" on page 149

Related tasks

″Configuring Secure Sockets Layer (SSL) support in a DB2 instance″ in Database Security Guide

″Configuring Secure Sockets Layer (SSL) support in non-Java DB2 clients″ in Database Security Guide

Related reference

″CATALOG TCPIP/TCPIP4/TCPIP6 NODE ″ in Command Reference

″security CLI/ODBC configuration keyword″ in Call Level Interface Guide and Reference, Volume 2

Transparent LDAP authentication and group lookup is supported (AIX)

DB2 Version 9.7 supports transparent LDAP on the AIX operating system which is a new option for implementing LDAP-based authentication and group lookup.

Transparent LDAP authentication and group lookup allows you to utilize LDAP-based authentication without having to first configure the DB2 environment using the DB2 LDAP security plug-ins. You can configure DB2 instances to authenticate users and acquire their groups through the AIX operating system. The AIX operating system will, in turn, perform the authentication and groups acquisition through an LDAP server. To enable transparent LDAP authentication, set the **DB2AUTH** registry variable to OSAUTHDB.

This support was first introduced in Version 9.5 Fix Pack 4.

Related concepts

″LDAP-based authentication and group lookup support″ in Database Security Guide

Passwords can be the maximum length supported by the operating system

In Version 9.7, when you configure the DB2 database manager to perform authentication by using the IBM operating system authentication plug-in, you can set the password length to the maximum length supported by the operating system.

Some operating systems might have further password rules, such as on minimum length and simplicity. Also you can configure some operating systems to use password encryption algorithms supplied by that operating system. For more information, see the applicable operating system documentation.

SYSMON authority has been extended to LIST commands and the db2mtrk command

To improve the database monitoring capability of a user holding system monitor (SYSMON) authority, SYSMON now includes the ability to run certain LIST commands. Also, SYSMON authority enables you to run the db2mtrk command to report memory pool allocation information.

The affected LIST commands are as follows:

- LIST DATABASE PARTITION GROUPS
- LIST DRDA INDOUBT TRANSACTIONS
- LIST PACKAGES
- LIST TABLES
- LIST TABLESPACE CONTAINERS
- LIST TABLESPACES
- LIST UTILITIES

Related reference

″LIST PACKAGES/TABLES ″ in Command Reference

″LIST TABLESPACE CONTAINERS ″ in Command Reference

″LIST TABLESPACES ″ in Command Reference

″LIST DRDA INDOUBT TRANSACTIONS ″ in Command Reference

″LIST DATABASE PARTITION GROUPS ″ in Command Reference

″db2mtrk - Memory tracker ″ in Command Reference

″LIST UTILITIES ″ in Command Reference

Chapter 11. Application development enhancements

Version 9.7 application development enhancements simplify database application development, improve application portability, and ease application deployment.

Database object management has been simplified as follows:

- Columns can be renamed using the ALTER TABLE statement (see “Columns can be renamed using the ALTER TABLE statement” on page 72).
- Most objects can be replaced during object creation using the OR REPLACE clause (see “REPLACE added as an option on several CREATE statements” on page 73).
- Unlimited REORG-recommended operations can be performed within a single transaction (see “Unlimited REORG-recommended operations can now be performed within a single transaction” on page 74).
- Some object can still be created even if errors occur during the compilation of their bodies (see “CREATE with errors support has been added for certain database objects” on page 74).
- Soft invalidation and automatic revalidation are supported for a subset of data definition language (DDL) statements (see “Soft invalidation and automatic revalidation support added” on page 75).
- Base table column types can be altered by using the ALTER TABLE statement in more cases (see “ALTER COLUMN SET DATA TYPE support has been extended” on page 77).
- The SELECT INTO statement supports the FOR UPDATE clause (see “The SELECT INTO statement supports the FOR UPDATE clause” on page 78).

SQL programming has been enhanced with this additional functionality:

- The TRUNCATE statement, which can quickly delete all rows from a database table (see “TRUNCATE added as an SQL statement” on page 78)
- Created temporary tables, which are a new type of user-defined temporary table (see “Created temporary tables are supported” on page 79)
- Numerous scalar functions additions and updates (see “Scalar function support has been extended” on page 79)
- Implicit casting, which automatically converts the data of one data type to another type based on an implied set of conversion rules (see “Implicit casting simplifies application enablement” on page 82)
- Choices for setting the precision of the fractional seconds of the TIMESTAMP data type (see “TIMESTAMP data type allows for parameterized precision” on page 82)
- LOB-type columns in temporary tables support (see “Temporary tables support LOB-type columns” on page 83)
- Public aliases (public synonyms) support (see “Public aliases simplify global object referencing” on page 83)
- For modules and sequences, private aliases within a schema support (see “Public aliases simplify global object referencing” on page 83)

Stored procedures development has been both simplified and enhanced as follows:

- Default values for parameters (see “Default values and named parameters promote flexibility for creating and calling procedures” on page 84)

- Argument assignment by parameter name when you call procedures (see “Default values and named parameters promote flexibility for creating and calling procedures” on page 84)
- Autonomous transactions, which commit work even if the invoking transaction itself is rolled back (see “Autonomous transactions are supported” on page 85)

General database application programming has been improved with the following additional enhancements:

- Extensions from Python applications to access IBM data servers (see “Python application development support has been added” on page 85)
- System-defined modules (see “System-defined modules simplify SQL PL and application logic” on page 86)
- An enhanced version of IBM Database Add-Ins for Visual Studio (see “IBM Database Add-Ins for Visual Studio have been enhanced” on page 88)
- Collection of common-signature and signature-stable stored procedures that are portable across IBM data servers (see “Common SQL API supports the development of portable administrative applications” on page 87)
- New DB2 sample programs as templates to create your own application programs (see “New DB2 sample programs have been added” on page 89)

SQL Procedural Language (SQL PL) programming has also been improved, see Chapter 12, “SQL Procedural Language (SQL PL) enhancements,” on page 107.

In addition, the IBM data server clients and drivers have been enhanced as follows:

- IBM Data Server Driver for JDBC and SQLJ includes support for Version 9.7 enhancements and includes additional features (see “JDBC and SQLJ support has been enhanced” on page 92).
- IBM Data Server Driver Package has been enhanced (see “ IBM Data Server Driver Package has been enhanced” on page 98).
- The IBM_DB Ruby driver, IBM PHP extensions, and IBM Data Server Provider for .NET support trusted contexts (see “Trusted context support has been extended” on page 99).
- Sysplex functionality is supported on IBM data server clients and non-Java data server drivers with a DB2 Connect license (see “Sysplex support is extended to IBM data server clients and non-Java data server drivers” on page 100).
- Call Level Interface (CLI) includes support for Version 9.7 enhancements and includes additional features (see “Call Level Interface (CLI) functionality has been enhanced” on page 101).
- The packaging of the data server drivers has been simplified (see “Component names have changed” on page 3).

Columns can be renamed using the ALTER TABLE statement

The RENAME COLUMN clause is a new option on the ALTER TABLE statement. You can now rename an existing column in a base table to a new name without losing stored data or affecting any privileges or label-based access control (LBAC) policies that are associated with the table.

Example

```
ALTER TABLE ORG RENAME COLUMN DEPTNUMB TO DEPTNUM
```

Related tasks

″Renaming tables and columns″ in Database Administration Concepts and Configuration Reference

Related reference

″ALTER TABLE ″ in SQL Reference, Volume 2

REPLACE added as an option on several CREATE statements

The OR REPLACE clause is a new option on the CREATE statement for several objects, including aliases, functions, modules, nicknames, procedures (including federated procedures), sequences, triggers, variables, and views. The object is replaced if it already exists; otherwise, it is created. This enhancement significantly reduces the effort required to change a database schema.

Privileges that were previously granted on an object are preserved when that object is replaced. In other respects, CREATE OR REPLACE is semantically similar to DROP followed by CREATE.

In the case of functions, procedures, and triggers, support applies to both inline objects and compiled objects. In the case of functions and procedures, support applies to both SQL and external functions and procedures.

If a module is replaced, all the objects within the module are dropped; the new version of the module contains no objects.

Example

Replace V1, a view that has dependent objects.

```
CREATE TABLE T1 (C1 INT, C2 INT);
CREATE TABLE T2 (C1 INT, C2 INT);

CREATE VIEW V1 AS SELECT * FROM T1;
CREATE VIEW V2 AS SELECT * FROM V1;

CREATE FUNCTION foo1()
 LANGUAGE SQL
 RETURNS INT
 RETURN SELECT C1 FROM V2;

CREATE OR REPLACE V1 AS SELECT * FROM T2;

SELECT * FROM V2;

VALUES foo1();
```

The replaced version of V1 references T2 instead of T1. Both V2 and foo1 are invalidated by the CREATE OR REPLACE statement. Under revalidation deferred semantics, `SELECT * FROM V2` successfully revalidates v2, but not foo1, which is revalidated by `VALUES foo1()`. Under revalidation immediate semantics, both V2 and foo1 are successfully revalidated by the CREATE OR REPLACE statement.

Related reference

″SQL statements″ in SQL Reference, Volume 2

″auto _reval - Automatic revalidation and invalidation configuration parameter″ in Database Administration Concepts and Configuration Reference

″Creating and maintaining database objects″ in Database Administration Concepts and Configuration Reference

Unlimited REORG-recommended operations can now be performed within a single transaction

Any ALTER TABLE operation that changes the format of data on disk requires a new version of the table descriptor, and is called a *REORG-recommended operation*. Such operations include dropping a column, altering a column type, or altering the nullability property of a column. An unlimited number of REORG-recommended operations per unit of work (UOW) is now permitted.

In previous versions, up to three ALTER TABLE statements of this type could execute against a table before reorganization was necessary. This limitation has been partially removed. A REORG TABLE command must be issued after a total of three units of work with REORG-recommended operations against the same table have occurred, and table reorganization is a prerequisite to any additional REORG-recommended operations against that table.

The row version is changed only once per unit of work when the first REORG-recommended operation is executed. Subsequent REORG-recommended operations in the same UOW do not create a new version of the row. As in previous versions, the data on disk is updated only if there is at least one drop column operation in subsequent ALTER TABLE statements.

Related reference

″ALTER TABLE ″ in SQL Reference, Volume 2

″REORG INDEXES/TABLE ″ in Command Reference

″Altering tables″ in Database Administration Concepts and Configuration Reference

CREATE with errors support has been added for certain database objects

Certain objects can now be created even if errors occur during the compilation of their bodies; for example, creating a view when the table to which it refers does not exist.

Such objects remain invalid until they are accessed. CREATE with errors support currently extends to views and inline SQL functions (not compiled functions). This feature is enabled if the **auto_reval** database configuration parameter is set to IMMEDIATE or DEFERRED_FORCE.

The errors that are tolerated during object creation are limited to the follow types:

- Any name resolution error, such as: a referenced table does not exist (SQLSTATE 42704, SQL0204N), a referenced column does not exist (SQLSTATE 42703, SQL0206N), or a referenced function cannot be found (SQLSTATE 42884, SQL0440N)
- Any nested revalidation failure. An object being created can reference invalid objects, and revalidation will be invoked for those invalid objects. If revalidation

of any referenced invalid object fails, the CREATE statement succeeds, and the created object will remain invalid until it is next accessed.

- Any authorization error (SQLSTATE 42501, SQL0551N)

An object can be created successfully even if there are multiple errors in its body. The warning message that is returned contains the name of the first undefined, invalid, or unauthorized object that was encountered during compilation. The SYSCAT.INVALIDOBJECTS catalog view contains information on invalid objects.

Example

```
CREATE VIEW V2 AS SELECT * FROM V1
```

If V1 does not exist, the CREATE VIEW statement completes successfully, but V2 remains invalid.

Related reference

″auto _reval - Automatic revalidation and invalidation configuration parameter″ in Database Administration Concepts and Configuration Reference

″Creating and maintaining database objects″ in Database Administration Concepts and Configuration Reference

Soft invalidation and automatic revalidation support added

Soft invalidation is a process whereby access to an object being invalidated is allowed to continue.

Previously, when an object was being altered or dropped, exclusive locking was used to ensure that all users of the object were no longer using it. This locking could result in applications waiting or rolling back because of deadlocks. Soft invalidation, however, avoids these waits, and allows any active access to continue, using the old version of the object. If the object has been recreated, any new access to the object will see the new version of the object; if the object has been dropped, new access to the object is not possible.

Soft invalidation is enabled through a new registry variable named **DB2_DDL_SOFT_INVAL**; by default, this registry variable is set to on. For more information, see “Miscellaneous variables”.

The following list shows the data definition language (DDL) statements for which soft invalidation is currently supported:

- CREATE OR REPLACE ALIAS
- CREATE OR REPLACE FUNCTION
- CREATE OR REPLACE TRIGGER
- CREATE OR REPLACE VIEW
- DROP ALIAS
- DROP FUNCTION
- DROP TRIGGER
- DROP VIEW

Soft invalidation support applies only to dynamic SQL and to scans done under the cursor stability (CS) and uncommitted read (UR) isolation levels.

In general, the database manager attempts to revalidate invalid objects the next time that those objects are used. However, if the **auto_reval** database configuration parameter is set to IMMEDIATE, invalid objects are revalidated immediately after they become invalid. For information about the dependent objects that are impacted when an object is dropped, and when those dependent objects are revalidated, see "DROP statement".

The following list shows the data definition language (DDL) statements for which automatic revalidation is currently supported:

- ALTER MODULE DROP FUNCTION
- ALTER MODULE DROP PROCEDURE
- ALTER MODULE DROP TYPE
- ALTER MODULE DROP VARIABLE
- ALTER NICKNAME (altering the local name or the local type)
- ALTER TABLE ALTER COLUMN
- ALTER TABLE DROP COLUMN
- ALTER TABLE RENAME COLUMN
- CREATE OR REPLACE ALIAS
- CREATE OR REPLACE FUNCTION
- CREATE OR REPLACE NICKNAME
- CREATE OR REPLACE PROCEDURE
- CREATE OR REPLACE SEQUENCE
- CREATE OR REPLACE TRIGGER
- CREATE OR REPLACE VARIABLE
- CREATE OR REPLACE VIEW
- DROP FUNCTION
- DROP NICKNAME
- DROP PROCEDURE
- DROP SEQUENCE
- DROP TABLE
- DROP TRIGGER
- DROP TYPE
- DROP VARIABLE
- DROP VIEW
- RENAME TABLE

Example

Disable soft invalidation support.

```
db2set DB2_DDL_SOFT_INVAL=off
```

Related concepts

″Soft invalidation of database objects″ in Database Administration Concepts and Configuration Reference

Related reference

″DROP ″ in SQL Reference, Volume 2

″SQL statements″ in SQL Reference, Volume 2

″auto _reval - Automatic revalidation and invalidation configuration parameter″ in Database Administration Concepts and Configuration Reference

ALTER COLUMN SET DATA TYPE support has been extended

The ALTER COLUMN SET DATA TYPE option on the ALTER TABLE statement has been extended to support all compatible types.

Previous support for the alteration of a base table column type (such as converting INTEGER to BIGINT or lengthening a LOB type column) was limited to changes that did not require a table scan. Column data types could not be cast to smaller types (for example, INTEGER to SMALLINT), and the full set of type alterations that are consistent with the cast functions for data types was not supported.

Altering the column data type can cause data loss. Some of this loss is consistent with casting rules; for example, blanks can be truncated from strings without returning an error, and converting a DECIMAL to an INTEGER results in truncation. To prevent unexpected errors, such as overflow errors, truncation errors, or any other kind of error returned by casting, existing column data is scanned, and messages about conflicting rows are written to the notification log. Column default values are also checked to ensure that they conform to the new data type.

If a data scan does not report any errors, the column type is set to the new data type, and the existing column data is cast to the new data type. If an error is reported, the ALTER TABLE statement fails.

Casting of VARCHAR, VARGRAPHIC, or LOB data types to types of lesser length is not supported.

Example

Change the data type of the SALES column in the SALES table from INTEGER to SMALLINT.

```
ALTER TABLE SALES ALTER COLUMN SALES SET DATA TYPE SMALLINT
DB20000I  The SQL command completed successfully.
```

Change the data type of the REGION column in the SALES table from VARCHAR(15) to VARCHAR(14).

```
ALTER TABLE SALES ALTER COLUMN REGION SET DATA TYPE VARCHAR(14)
...
SQL0190N  ALTER TABLE "ADMINISTRATOR.SALES" specified attributes for column
"REGION" that are not compatible with the existing column.  SQLSTATE=42837
```

Change a column type in a base table. There are views and functions that are directly or indirectly dependent on the base table.

```
CREATE TABLE T1 (C1 INT, C2 INT);

CREATE VIEW v1 AS SELECT C1, C2 FROM T1;
CREATE VIEW v2 AS SELECT C1, C2 FROM V1;

CREATE FUNCTION foo1 ()
 LANGUAGE SQL
 RETURNS INT
 RETURN SELECT C2 FROM T1;

CREATE VIEW v3 AS SELECT C2 FROM V2
 WHERE C2 = foo1();

CREATE FUNCTION foo2 ()
 LANGUAGE SQL
 RETURNS INT
 RETURN SELECT C2 FROM V3;

 ALTER TABLE T1
 ALTER COLUMN C1
  SET DATA TYPE SMALLINT;

SELECT * FROM V2;
```

The ALTER TABLE statement, which down casts the column type from INTEGER to SMALLINT, invalidates V1, V2, V3, and foo2. Under revalidation deferred semantics, `SELECT * FROM V2` successfully revalidates V1 and V2, and the C1 columns in both V1 and V2 are changed to SMALLINT. But V3 and foo2 are not revalidated, because they are not referenced after being invalidated, and they are above V2 in the dependency hierarchy chain. Under revalidation immediate semantics, the ALTER TABLE statement revalidates all the dependent objects successfully.

Related reference

″ALTER TABLE ″ in SQL Reference, Volume 2

″Casting between data types″ in SQL Reference, Volume 1

″Altering tables″ in Database Administration Concepts and Configuration Reference

The SELECT INTO statement supports the FOR UPDATE clause

You can use the optional FOR UPDATE clause in the SELECT INTO statement to transfer into DB2 environments those applications from other database vendors that use that feature. Many applications use this feature to fetch a single row and then update that row by a searched update.

Related reference

″SELECT INTO ″ in SQL Reference, Volume 2

TRUNCATE added as an SQL statement

Version 9.7 includes a new TRUNCATE statement that you can use to quickly delete all rows from a database table.

Unlike the DELETE statement, the TRUNCATE statement cannot be rolled back and the keyword IMMEDIATE is mandatory to indicate this fact.

Related reference

″DB2_COMPATIBILITY_VECTOR registry variable″ in SQL Procedural Languages: Application Enablement and Support

″TRUNCATE ″ in SQL Reference, Volume 2

Created temporary tables are supported

Created temporary tables are a new type of user-defined temporary table introduced in DB2 Version 9.7. Prior to this release, the declared temporary table was the only type of user-defined temporary table.

As with a declared temporary table, an application session can use a created temporary table to store intermediate result sets for manipulation or repeated references without interfering with concurrently running applications. The main difference between a declared temporary table and a created temporary table is that the definition of a created temporary table is stored persistently in the DB2 catalog. In other words, although the contents of a created temporary table is private to a session, its definition is shared across all concurrent sessions. The persistent storage of the created temporary table definition results in the following operational differences:

- After an application session defines a created temporary table, concurrently running sessions do not have to redefine it.
- You can reference a created temporary table in SQL functions, triggers, and views.

Also, any connection can refer to a created temporary table at any time without the need for a setup script to initialize the created temporary table. A connection can access only the rows that it inserts.

Other DB2 family products, such as DB2 for z/OS®, and the SQL Standard support created temporary tables.

You can use the cgtt.db2 and the Cgtt.java sample programs to learn how to use this feature.

Related concepts

“New DB2 sample programs have been added” on page 89

Related reference

″CREATE GLOBAL TEMPORARY TABLE ″ in SQL Reference, Volume 2

Scalar function support has been extended

Function support has been extended by the addition of several new scalar functions and enhancements to existing scalar functions.

These functions provide built-in support for functionality that would otherwise have to be developed through user-defined functions (UDFs). Moreover, built-in functions are superior to UDFs in terms of performance.

There are sample scripts provided to show you how to use the new scalar functions.

New scalar functions

ADD_MONTHS
Returns a datetime value that represents an expression plus a specified number of months.

ARRAY_DELETE
Deletes elements from an array.

ARRAY_FIRST
Returns the smallest array index value of the array.

ARRAY_LAST
Returns the largest array index value of the array.

ARRAY_NEXT
Returns the next larger array index value for an array relative to the specified array index argument.

ARRAY_PRIOR
Returns the next smaller array index value for an array relative to the specified array index argument.

CURSOR_ROWCOUNT
Returns the cumulative count of all rows fetched by the specified cursor since the cursor was opened.

DAYNAME
Returns a character string containing the name of the day (for example, Friday).

DECFLOAT_FORMAT (TO_NUMBER is a synonym of this scalar function)
Converts a string to a DECFLOAT data type.

EXTRACT
Returns a portion of a date or timestamp.

INSTR (which is a synonym for the LOCATE_IN_STRING scalar function)
Returns the starting position of a string within another string.

INITCAP
Takes a string expression and returns a string expression with the first character of each *word* in uppercase, and all other letters in lowercase.

LAST_DAY
Returns a datetime value that represents the last day of the month.

LOCATE_IN_STRING (INSTR is a synonym of this scalar function)
Returns the starting position of a string within another string.

LPAD Adds characters, symbols, or spaces to the left side of a string.

MONTHNAME
Returns a character string containing the name of the month (for example, January).

MONTHS_BETWEEN
Returns an estimate of the number of months between two expressions.

NEXT_DAY
Returns a datetime value that represents the first weekday later than the date in a specified expression.

ROUND_TIMESTAMP
Returns a timestamp value from an expression rounded to a specified unit.

RPAD Adds characters, symbols, or spaces to the right side of a string.

TO_CLOB (which is a synonym for the CLOB scalar function)
Converts character data to the CLOB data type.

TO_NUMBER (which is a synonym for the DECFLOAT_FORMAT scalar function)
Returns a DECFLOAT(34) value that is based on the interpretation of a string using the specified format.

TO_TIMESTAMP (which is a synonym for the TIMESTAMP_FORMAT scalar function)
Returns a timestamp that is based on the interpretation of a string using the specified format.

TRIM_ARRAY
Deletes elements from the end of an array.

TRUNC_TIMESTAMP
Adds characters, symbols, or spaces to the right side of a string.

Updated scalar functions

GRAPHIC
Converts input data to the GRAPHIC data type. Support for additional input data types has been added.

TIMESTAMP_FORMAT (TO_DATE and TO_TIMESTAMP are synonyms of this scalar function)
Returns a timestamp that is based on the interpretation of the input string using the specified format. Support for locale sensitive format options has been added.

TO_CHAR (which is a synonym for the VARCHAR_FORMAT scalar function)
Returns a string that is based on the interpretation of the input string using the specified format. Support for locale sensitive format options has been added.

TO_DATE (which is a synonym for the TIMESTAMP_FORMAT scalar function)
Returns a timestamp that is based on the interpretation of the input string using the specified format. Support for locale sensitive format options has been added.

VARCHAR_FORMAT (TO_CHAR is a synonym of this scalar function)
Returns a string that is based on the interpretation of the input string using the specified format. Support for locale sensitive format options has been added.

VARGRAPHIC
Converts input data to the VARGRAPHIC data type. Support for additional input data types has been added.

Related concepts

“New DB2 sample programs have been added” on page 89

″Introduction to DB2 compatibility features″ in SQL Procedural Languages: Application Enablement and Support

Implicit casting simplifies application enablement

Version 9.7 introduces support for implicit casting. Implicit casting is the automatic conversion of data of one data type to data of another data type based on an implied set of conversion rules. This automatic conversion occurs in support of weak typing.

Prior to Version 9.7, strong typing was used during comparisons and assignments. Strong typing requires matching data types, which means that you must explicitly convert one or both data types to a common data type before performing comparisons or assignments.

In Version 9.7, the rules used during comparisons and assignments have been relaxed. If two objects have mismatched types, implicit casting is used to perform comparisons or assignments if a reasonable interpretation of the data types can be made. Implicit casting is also supported during function resolution. When the data types of the arguments of a function being invoked cannot be promoted to the data types of the parameters of the selected function, the data types of the arguments are implicitly cast to the data types of the parameters. For more information, see “Functions”.

Implicit casting reduces the amount of SQL statements that you must modify when enabling applications that run on data servers other than DB2 data servers to run on DB2 Version 9.7. In many cases, you no longer have to explicitly cast data types when comparing or assigning values with mismatched data types.

You can use the implicitcasting.db2 and the ImplicitCasting.java sample programs to learn how to use this feature.

Version 9.7 includes another enhancement that enables you to use untyped parameter markers and untyped NULL keywords almost anywhere in an SQL statement where you can use an expression. For more information, see “Determining data types of untyped expressions”.

Related concepts

“New DB2 sample programs have been added” on page 89

Related reference

″Functions″ in SQL Reference, Volume 1

″Casting between data types″ in SQL Reference, Volume 1

″Assignments and comparisons″ in SQL Reference, Volume 1

″Rules for result data types″ in SQL Reference, Volume 1

″Determining data types of untyped expressions″ in SQL Reference, Volume 1

TIMESTAMP data type allows for parameterized precision

The TIMESTAMP data type is now parameterized to control the precision of the fractional seconds. The range is 0 (no fractional seconds) to 12 (picoseconds). Using the TIMESTAMP data type without a parameter continues to be supported as in previous releases with the default precision of 6 (microseconds).

There are two benefits in having a parameterized TIMESTAMP data type:

- An increased maximum precision of 12 improves the granularity of the TIMESTAMP data type.
- You can control the precision to meet the needs of your application and use only the storage required to meet those needs. For example, for an application that needs only the date and time, you can now specify a precision of 0, which saves 3 bytes per TIMESTAMP compared to the storage that was required in previous releases.

Related reference

″Constants″ in SQL Reference, Volume 1

″Datetime values″ in SQL Reference, Volume 1

″Assignments and comparisons″ in SQL Reference, Volume 1

″Rules for result data types″ in SQL Reference, Volume 1

″Datetime operations and durations″ in SQL Reference, Volume 1

Temporary tables support LOB-type columns

In Version 9.7, declared temporary tables can contain LOB-type columns. LOB-type columns are also supported for created temporary tables.

Prior to Version 9.7, you could not store LOB data in declared temporary tables, so applications either had to work around not having the LOB data in a declared temporary table or had to use a regular table.

For both types of user-defined temporary tables, the values for LOB-type columns are stored in the same table space in which the temporary table is instantiated.

Related reference

″DECLARE GLOBAL TEMPORARY TABLE ″ in SQL Reference, Volume 2

″CREATE GLOBAL TEMPORARY TABLE ″ in SQL Reference, Volume 2

Public aliases simplify global object referencing

Referencing objects outside the current schema has been globalized in Version 9.7 by the use of public aliases, which are also known as public synonyms. Also, support for aliases (sometimes referred to as private aliases) within a schema has been extended.

Prior to this release, you could define aliases for another alias, a nickname, a table, or a view. You can now also define aliases for modules and sequences.

You can define public aliases for all of the objects that you can define private aliases for, namely, another alias (private or public), a nickname, a module, a sequence, a table, or a view. A public alias is implicitly qualified with SYSPUBLIC, and these aliases can be referenced using an unqualified or a qualified name by any authorization ID.

The benefits of public aliases are as follows:

- An object can be referenced independent of the current SQL path or CURRENT SCHEMA setting by its simpler, one-part name.
- You can use a single DDL statement to expose an object to all users of a database, which reduces the amount of typing needed.

You can use the public_alias.db2 sample program to learn how to use this feature.

Related concepts

"New DB2 sample programs have been added" on page 89

Related reference

″CREATE ALIAS ″ in SQL Reference, Volume 2

Default values and named parameters promote flexibility for creating and calling procedures

Starting in Version 9.7, you can create procedures and specify default values for parameters. When calling procedures, arguments can be assigned to parameters by name, allowing you to pass named arguments in any order.

With default parameter values defined in the procedure, and the ability to invoke a procedure with named arguments, you now have the following options when calling the procedure:

- Explicitly use the name of the parameter to name the argument
- Omit argument values such that the default value will be assigned
- Specify named arguments in any order

These enhancements allow for replacing an existing procedure with added parameters using defaults such that existing invocations of the procedure do not need to be changed.

You can use the defaultparam.db2 sample program to learn how to use this feature.

Examples

- *Example 1*: Create a procedure FOO with 3 parameters (**parm1**, **parm2** and **parm3**) and assign default values for each parameter.

  ```
  CREATE PROCEDURE FOO(parm1 INT DEFAULT -1, parm2 INT DEFAULT -2, parm3 INT DEFAULT -3)
  ```

- *Example 2*: Invoke the FOO procedure by explicitly including parameter names.

  ```
  CALL FOO (parm1=>10, parm2=>20, parm3=>30)
  ```

  The invocation of FOO is identical to FOO (10, 20, 30).

- *Example 3*: Invoke the FOO procedure by omitting argument values.

  ```
  CALL FOO (40)
  ```

  FOO is invoked using a value of 40 for **parm1** and uses the default values of -2 for **parm2** and of -3 for **parm3**.

- *Example 4*: Invoke the FOO procedure by passing arguments to the named parameters in any order.

  ```
  CALL FOO (parm3=>30, parm2=>20)
  ```

  FOO is invoked using the default value of -1 for **parm1**, the value 20 for **parm2** and the value 30 for **parm3**.

- *Example 5*: Invoke the FOO procedure by passing a mixture of named and unnamed parameters. Values that are not passed in by name are mapped according to their position in the invocation.

  ```
  CALL FOO (40, parm3=>10)
  ```

  FOO is invoked using a value of 40 for **parm1**, the default value of -2 for **parm2**, and a value of 10 for **parm3**.

Related concepts

“New DB2 sample programs have been added” on page 89

Related reference

″CALL ″ in SQL Reference, Volume 2

″CREATE PROCEDURE (external) ″ in SQL Reference, Volume 2

″CREATE PROCEDURE (SQL) ″ in SQL Reference, Volume 2

″CREATE PROCEDURE statement (PL/SQL)″ in SQL Procedural Languages: Application Enablement and Support

Autonomous transactions are supported

Version 9.7 provides a mechanism to run and commit a block of statements independent of the outcome of the invoking transaction. This mechanism, referred to as an *autonomous transaction*, implies that work is committed even if the invoking transaction itself is rolled back.

This feature is particularly useful when you migrate applications using autonomous features supported by other database systems to Version 9.7. You can migrate those applications more easily.

To enable an autonomous transaction, specify the new keyword AUTONOMOUS when using the CREATE PROCEDURE statement. A procedure that you define with this new keyword runs within its own session, meaning that the procedure is independent of the calling procedure. If an autonomous procedure is successfully completed, the work is committed; if the procedure fails, the work is rolled back. In either case, the calling transaction is not affected.

You can use the autonomous_transaction.db2 sample program to learn how to use this feature.

Related concepts

“New DB2 sample programs have been added” on page 89

Related reference

″CREATE PROCEDURE ″ in SQL Reference, Volume 2

Python application development support has been added

Extensions are available for accessing IBM data server databases from a Python application.

The following extensions are available:

ibm_db API
: Provides the best support for advanced features, including pureXML support and access to metadata

ibm_db_dbi API
: Implements Python Database API Specification v2.0, which provides basic functions for interacting with databases but does not offer the advanced features provided by ibm_db

ibm_db_sa adaptor
: Provides support for using SQLAlchemy to access IBM data servers

These extensions allow Python applications to access the following IBM data servers:

- DB2 Database for Linux, UNIX, and Windows, Version 9.1 Fix Pack 2 (and later)
- DB2 UDB for Linux, UNIX, and Windows, Version 8 Fixpak 15 (and later)
- Remote connections to IBM DB2 Universal Database on i5/OS® V5R3, with PTF SI27358 (includes SI27250)
- Remote connections to DB2 for IBM i 5.4 (and later) with PTF SI27256
- Remote connections to DB2 for z/OS, Version 8 and Version 9
- IBM Informix® Dynamic Server, Version 11.10 (and later)

Related concepts

″Python and SQLAlchemy application development for IBM data servers″ in Getting Started with Database Application Development

Related reference

″Python downloads and related resources″ in Developing Perl, PHP, Python, and Ruby on Rails Applications

System-defined modules simplify SQL PL and application logic

New system-defined modules provide an easy-to-use programmatic interface for performing a variety of application development tasks. A large number of new, pre-written, system defined functions and modules are included in Version 9.7.

The following modules contain routines and procedures that provide enhanced capabilities for communicating through messages and alerts, for creating, scheduling, and managing jobs, for operating on large objects, for executing dynamic SQL, for working with files on the database server file system, and for sending email.

Table 6. Supported modules

Module	Description
DBMS_ALERT module	Provides a set of procedures for registering for alerts, sending alerts, and receiving alerts.
DBMS_JOB module	Provides a set of procedures for creating, scheduling, and managing jobs. DBMS_JOB is an alternate interface for the Administrative Task Scheduler (ATS).
DBMS_LOB module	Provides a set of routines for operating on large objects.
DBMS_OUTPUT module	Provides a set of procedures for putting messages (lines of text) in a message buffer and getting messages from the message buffer within a single session. These procedures are useful during application debugging when you need to write messages to standard output.
DBMS_PIPE module	Provides a set of routines for sending messages through a pipe within or between sessions that are connected to the same database.
DBMS_SQL module	Provides a set of procedures for executing dynamic SQL.
DBMS_UTILITY module	Provides a set of utility routines.

Table 6. Supported modules (continued)

Module	Description
UTL_DIR module	Provides a set of routines for maintaining directory aliases that are used with the UTL_FILE module.
UTL_FILE module	Provides a set of routines for reading from and writing to files on the database server file system.
UTL_MAIL module	Provides a set of procedures for sending email.
UTL_SMTP module	Provides a set of routines for sending e-mail using the Simple Mail Transfer Protocol (SMTP).

Common SQL API supports the development of portable administrative applications

The common SQL API provides a collection of common-signature and signature-stable stored procedures that are portable across IBM data servers. You can use these stored procedures to create applications that perform common administrative functions such as getting and setting the values of configuration parameters or getting system and error information.

IBM data servers have many ways to obtain data for administrative purposes and to issue administrative commands. Prior to the introduction of this feature, the variety of methods for performing administrative functions, their different syntaxes, and their security options resulted in tight coupling between tool and data server versions, high implementation complexity on the tools side, and slow integration.

The common SQL API addresses these problems by providing the following benefits:

A single access method
The stored procedures are accessed through SQL.

A simple security model
The stored procedures require only EXECUTE privilege, with no additional dependencies.

The ability to add additional stored procedures in fix packs
The set of stored procedures can be expanded in future fix packs to provide support for performing additional administrative functions.

Data server version independence
The stored procedures provide syntactically identical XML parameters and error handling across all data servers to ensure data server version independence. Signature stability and commonality are achieved by using simple XML documents (with a common DTD) as parameters. Version, platform, and technology differences are expressed through different key value pairs in hierarchical property lists.

The ability for clients to determine supported features
Clients can call the stored procedures to determine the highest supported versions.

Support for automation

You can use the stored procedures in automated scripts.

The common SQL API currently provides the following stored procedures.

Table 7. Common SQL API stored procedures

Procedure name	Description
CANCEL_WORK	Cancels either a specific activity (for example, an SQL statement) or all activities for a connected application
GET_CONFIG	Retrieves data server configuration data, including nodes.cfg file data, database manager configuration data, database configuration data, and DB2 registry settings from all database partitions
GET_MESSAGE	Retrieves the short message text, long message text, and SQLSTATE for an SQLCODE
GET_SYSTEM_INFO	Retrieves information about the data server, including information about the system, the current instance, installed DB2 products, environment variables, and available processors
SET_CONFIG	Updates the configuration parameters retrieved by the GET_CONFIG procedure

Related concepts

″Common SQL API procedures″ in Administrative Routines and Views

IBM Database Add-Ins for Visual Studio have been enhanced

IBM Database Add-Ins for Visual Studio, which provide tools for rapid application development, database schema development, and debugging, offer even better support in Version 9.7.

You can use the following enhancements:

- You can install IBM Database Add-Ins for Visual Studio in Visual Studio 2008 in addition to the existing Visual Studio 2005 support.
- The add-ins now support IBM DB2 Version 9.7 for Linux, UNIX, and Windows and UniVerse® 10.3. Availability of some features and data type compatibility depends on the data server that you use.
- You can use a feature in the add-ins with IBM Optim™ pureQuery Runtime to capture SQL statements in your .NET applications. You can then run the SQL statements statically. Using static SQL avoids the need to prepare certain statements at run time and can improve security and performance of your applications.
- You can define compound statements in your DB2 SQL procedures.
- You can use trusted contexts during data connections.
- You can choose to use Internet Protocol version 6 (IPv6) format for server addresses in data connections.
- You can use the following Microsoft® Visual Studio 2008 features:
  - Windows Presentation Foundation (WPF). You can generate XAML-type applications from database tables, views, and procedures.

  - Windows Workflow Foundation (WF) integration with Windows Communication foundation (WCF). You can drag an IBM DB2 custom activity from the WF project toolbox to the Activity Designer.
  - ASP.NET AJAX. You can call ASP.NET Web services from the browser by using client script.
- You can use localhost data connections without specifying user IDs and passwords.
- You can group IBM database objects by schemas in the Server Explorer.

In addition, the add-ins contain additional V9.7 features that enhance working with other IBM data servers.

Related concepts

″DB2 integration in Visual Studio″ in Developing ADO.NET and OLE DB Applications

New DB2 sample programs have been added

You can use DB2 sample programs as templates to create your own application programs and to learn about DB2 product functionality.

The samples are bundled with all of the server editions of the DB2 Database for Linux, UNIX, and Windows and with the IBM Data Server Client. You can find the samples in the following locations:

- On Windows operating systems: *%DB2PATH%*\sqllib\samples (where *%DB2PATH%* is the directory where the DB2 product is installed, as set by the **DB2PATH** environment variable)
- On UNIX operating systems: *$HOME*/sqllib/samples (where *$HOME* is the home directory of the instance owner, as set by the **$HOME** environment variable)

The new sample programs are as follows.

Table 8. New Version 9.7 sample programs

Category	Feature or enhancement	Sample description
Application development	Currently committed semantics for improved concurrency	The AIRLINE.war sample program shows how to use currently committed semantics with the cursor stability isolation level to avoid lock waits and deadlock scenarios.
	Improved scalar functions	The scalarfunction.db2 and ScalarFunctions.java sample programs show how to use built-in support for functionality that would otherwise be developed by using user-defined functions (UDFs).
	Implicit casting	The implicitcasting.db2 and ImplicitCasting.java sample programs show how to use implicit casting with data type assignments, comparisons, and NULL values.
	Created temporary tables	The cgtt.db2 and Cgtt.java sample programs show how to store the intermediate results and how to use created temporary tables with procedures, functions, triggers, and views.
	Autonomous transactions	The autonomous_transaction.db2 sample program shows how to track events when a procedure created by using the AUTONOMOUS keyword of the CREATE PROCEDURE statement attempts to access restricted data.
	Default parameters	The defaultparam.db2 sample program shows how to use the keyword **DEFAULT** as a parameter in CREATE PROCEDURE and CALL statements.
	Public aliases	The public_alias.db2 sample program shows how to use public aliases for database objects such as tables and modules.
	Date compatibility features	The datecompat.db2 sample program shows DATE formats, DATE addition and subtraction, scalar functions, and interpretation of DATE data type as TIMESTAMP (0) data type in date compatibility mode.
	Modules	The modules.db2 sample program shows how to create modules, public and private module objects such as procedures and functions; user-defined row data types; associative arrays; parameterized cursors, and strongly-typed and weakly-typed cursors. The program also shows how to pass the data types and cursors as parameters to the procedures and functions and how to use the data types and cursors within the procedures and functions. The program also shows how SQL PL supports triggers, UDFs and stand-alone compiled compound statements.

Table 8. New Version 9.7 sample programs (continued)

Category	Feature or enhancement	Sample description
XML	Partitioned tables	The xrpart.db2 sample program shows how to use XML in partitioned tables and how these tables support global indexes.
	Partitioned environments	The xmlpartition.db2 sample program shows how to use XML in partitioned databases, MDC, and partitioned tables.
	XML support for MDC tables	The xmlmdc.db2 and XmlMdc.java sample programs show how to move data from non-MDC tables to MDC tables. The program also highlights the use of block indexes, XML indexes, and faster insertion and deletion.
	Improved XML data type support	The XmlUdfs.java, xmludfs.db2, xmludfs.sqc, and xmludfs.c sample programs show how to use the XML data type. The programs show how to pass the XML data type as an input parameter, declare XML data type local variables, and return values when using scalar functions, sourced functions, SQL-bodied UDFs and table UDFs.
	Inline LOB functions	The xmldbafn.db2 sample program shows how to use the ADMIN_IS_INLINED function to determine whether all XML documents are inlined. The program also shows how to use the ADMIN_EST_INLINE_LENGTH function to get the estimated inline length for XML documents that are not inlined.
	Annotated XML schema decomposition	The xmldecomposition.db2, XmlDecomposition.java, and xmldecomposition.sqc sample programs show how to decompose instance documents and use annotated XML decomposition to store the data in relational tables.
	Index reorganization	The xmlolic.db2 sample program shows how to use the REORG command with the **ALLOW WRITE ACCESS** parameter to reorganize indexes defined on a table and how to reorganize nonpartitioned indexes on a partitioned table.
	Declared global temporary tables with XML	The xmlindgtt.db2 sample program shows how to use declared global temporary tables with the XML data type and how to create indexes on XML documents.
Monitoring	Java-based tool for event monitor reports	The db2evmonfmt.java sample program shows how to produce readable flat-text output or formatted XML output from the data generated by an event monitor that uses an unformatted event table.

Table 8. New Version 9.7 sample programs (continued)

Category	Feature or enhancement	Sample description
Workload management	DB2 workload manager (WLM) tiered service class configuration setup	The wlmtiersdefault.db2 sample program shows how to set up a DB2 WLM tiered service class configuration for a database. The database throughput in response to the elapsed execution time of incoming activities can be improved using this tiered service. This sample program also demonstrates the use of service classes, workloads, and thresholds.
	DB2 WLM tiered service class configuration setup	The wlmtierstimerons.db2 script shows how to set up a DB2 WLM manager tiered service class configuration for a database. Unlike the wlmtiersdefault.db2 script, the wlmtierstimerons.db2 script takes the estimated cost into account when initially mapping DML activities to service classes.
	Dropping of DB2 WLM service classes, thresholds, workloads, work classes, and work action sets	The wlmtiersdrop.db2 script shows how to drop all service classes, thresholds, workloads, work classes, and work action sets that were created by the wlmtiersdefault.db2 and wlmtierstimerons.db2 scripts.

IBM data server clients and drivers enhancements

Some IBM data server clients and drivers have been enhanced with new and improved features that improve application performance and reliability.

To make use of Version 9.7 features, you must upgrade to a Version 9.7 IBM data server client or driver.

In general, you can use Version 9.1 and Version 9.5 clients and drivers for running applications, developing applications, and performing database administration tasks on DB2 Version 9.7. Likewise, you can use Version 9.7 clients and drivers for running applications, developing applications, and performing administrative tasks on DB2 Version 9.5 and Version 9.1 servers. However, the functionality available can differ based on the combination of the version levels used on the server and client (or driver).

JDBC and SQLJ support has been enhanced

The IBM Data Server Driver for JDBC and SQLJ contains a number of major enhancements for Version 9.7.

Those enhancements are available in version 3.57 or version 4.7 of the driver. Version 3.57 contains JDBC 3.0 and earlier functions, and version 4.7 contains JDBC 4.0 and earlier functions.

Named parameter markers support

The IBM Data Server Driver for JDBC and SQLJ adds the following new methods to support named parameter markers.

- DB2PreparedStatement methods for assigning a value to a named parameter marker:
  - setJccArrayAtName
  - setJccAsciiStreamAtName (Some forms of this method require version 4.7 of the driver.)
  - setJccBigDecimalAtName

  - setJccBinaryStreamAtName (Some forms of this method require version 4.7 of the driver.)
  - setJccBlobAtName (Some forms of this method require version 4.7 of the driver.)
  - setJccBooleanAtName
  - setJccByteAtName
  - setJccBytesAtName
  - setJccCharacterStreamAtName (Some forms of this method require version 4.7 of the driver.)
  - setJccClobAtName (Some forms of this method require version 4.7 of the driver.)
  - setJccDateAtName
  - setJccDoubleAtName
  - setJccFloatAtName
  - setJccIntAtName
  - setJccLongAtName
  - setJccNullAtName
  - setJccObjectAtName
  - setJccShortAtName
  - setJccSQLXMLAtName (This method requires version 4.7 of the driver.)
  - setJccStringAtName
  - setJccTimeAtName
  - setJccTimestampAtName
  - setJccUnicodeStreamAtName
- DB2CallableStatement method for registering named parameter markers as stored procedure OUT parameters:
  - registerJccOutParameterAtName

Parameter names in JDBC and SQLJ stored procedure calls support

In previous versions of the IBM Data Server Driver for JDBC and SQLJ, only forms of CallableStatement.RegisterOutParameter, CallableStatement.setXXX, and CallableStatement.getXXX methods that used *parameterIndex* were supported. With versions 3.57 and 4.7 of the driver, *parameterName* is also supported in those methods. *parameterName* is a name that is specified for a parameter in the stored procedure definition.

Alternatively, for JDBC applications, new syntax allows the application to map parameter markers in the CALL statement to the parameter names in the stored procedure definition. For example, in a JDBC application, `CALL MYPROC (A=>?)` maps a parameter marker to stored procedure parameter A.

For SQLJ applications, new syntax allows the application to map host variable names in the CALL statement to the parameter names in the stored procedure definition. For example, in an SQLJ application, `CALL MYPROC (A=>:INOUT x)` maps host variable x to stored procedure parameter A.

With the new syntax, you do not need to specify all parameters in the CALL statement. Unspecified parameters take the default values that are specified in the stored procedure definition.

Access to currently committed data support

The IBM Data Server Driver for JDBC and SQLJ adds the concurrentAccessResolution Connection or DataSource property. That property specifies whether the IBM Data Server Driver for JDBC and SQLJ requests that a read transaction can access a committed and consistent image of rows that are incompatibly locked by write transactions, if the data source supports accessing currently committed data, and the application isolation level is cursor stability (CS) or read stability (RS).

In addition, the IBM Data Server Driver for JDBC and SQLJ adds the DB2Connection.setDBConcurrentAccessResolution method, which lets you override the concurrentAccessResolution setting for new statements that are created on an existing Connection. The driver also provides the DB2Connection.getDBConcurrentAccessResolution method, which lets you check the concurrent access resolution setting.

Stored procedures calls with cursor type parameters support

Stored procedures that are created on DB2 Database for Linux, UNIX, and Windows can have OUT parameters of the cursor type. The IBM Data Server Driver for JDBC and SQLJ supports retrieval of data from OUT parameters of the cursor type in JDBC and SQLJ applications. For registration of OUT parameters with the cursor type, the IBM Data Server Driver for JDBC and SQLJ adds the DB2Types.CURSOR data type.

Statement concentrator support

DB2 Database for Linux, UNIX, and Windows statement concentrator support is the ability to bypass preparation of a statement when it is the same as a statement in the dynamic statement cache, except for literal values. If statement concentrator support is enabled on a DB2 Database for Linux, UNIX, and Windows data source, you can use the statementConcentrator Connection or DataSource property to specify whether the IBM Data Server Driver for JDBC and SQLJ uses the statement concentrator support.

In addition, the IBM Data Server Driver for JDBC and SQLJ adds the DB2Connection.setDBStatementConcentrator method, which lets you override the statementConcentrator setting for new statements that are created on an existing Connection. The driver also provides the DB2Connection.getDBStatementConcentrator method, which lets you check the statement concentrator setting.

Variable-length timestamp support

DB2 Database for Linux, UNIX, and Windows supports timestamp columns of the form TIMESTAMP(*p*), where the precision of the timestamp value, *p*, is between 0 and 12. The IBM Data Server Driver for JDBC and SQLJ adds support for update and retrieval of values in a TIMESTAMP(*p*) column in JDBC and SQLJ applications.

The maximum precision of a Java timestamp value is 9, so there can be a loss of precision during data retrieval if *p*>9.

Retrieving special register settings support

For connections to DB2 for z/OS Version 8 or later, DB2 Database for Linux, UNIX, and Windows Version 8 or later, or DB2 UDB for iSeries® V5R3 or later, the IBM Data Server Driver for JDBC and SQLJ adds the DB2Connection.getJccSpecialRegisterProperties method. This method lets you retrieve the current special register settings for the data source, for special registers that the IBM Data Server Driver for JDBC and SQLJ supports.

Choice of output formats for DECIMAL or DECFLOAT data support

The decimalStringFormat Connection or DataSource property lets you choose the string format in which data from a DECIMAL or DECFLOAT column is retrieved, for an application that runs with the SDK for Java Version 1.5 or later. You can retrieve the data in the format that java.math.BigDecimal.toString method uses, which is the default. Alternatively, you can retrieve the data in the format that java.math.BigDecimal.toPlainString uses.

Compound SQL statements support

SQLJ statement clauses in SQLJ applications or SQL statements in JDBC applications can now include compound statements. A compound statement is a BEGIN-END block that includes SQL and procedural statements. All compound statements are executed dynamically, including those in SQLJ applications.

Savepoints support

The IBM Data Server Driver for JDBC and SQLJ supports setting of savepoints for connections to IBM Informix Dynamic Server data servers.

Batch insert operations support

The IBM Data Server Driver for JDBC and SQLJ adds the atomicMultiRowInsert Connection or DataSource property for connections to DB2 Database for Linux, UNIX, and Windows Version 8 and later data servers, DB2 for z/OS Version 8 and later data servers, or IBM Informix Dynamic Server V11.10 and later data servers. For connections to DB2 for z/OS, atomicMultiRowInsert is not supported for SQLJ applications. The atomicMultiRowInsert property lets you specify whether batch insert operations that use the PreparedStatement interface have atomic or non-atomic behavior. Atomic behavior means that a batch operation succeeds only if all insert operations in the batch succeed. Non-atomic behavior, which is the default, means that insert operations succeed or fail individually.

Implicit closing of result sets enhancements

The queryCloseImplicit Connection or DataSource property specifies whether cursors are closed immediately after all rows are fetched. A new value of QUERY_CLOSE_IMPLICIT_COMMIT (3) is added, to specify that cursors are closed after all rows are fetched, and in addition, if the application is in autocommit mode, a commit request is sent to the data source.

Diagnostics for binding of SQLJ applications enhancements

When an SQLJ application is bound, and an SQL error or warning occurs, the following new diagnostic information is returned:

- The SQL statement
- The line number in the program of the SQL statement
- The error or warning code and the SQLSTATE value
- The error message

Client reroute enhancements

Client reroute support is enhanced in the following ways:

- Seamless failover is added to client reroute operation.

  During client reroute, if a connection is in a clean state, you can use the enableSeamlessFailover property to suppress the SQLException with error code -4498 that the IBM Data Server Driver for JDBC and SQLJ issues to indicate that a failed connection was re-established.
- Client affinities are added to cascaded failover support.

  For cascaded failover, you can use the enableClientAffinitiesList property to control the order in which primary and alternate server reconnections are attempted after a connection failure.

Statement.setMaxRows performance enhancements

For connections to DB2 for z/OS servers, the Statement.setMaxRows method has been modified to provide better performance.

Connections to IDS enhancements

For connections to IDS servers, the following enhancements are added:

- IDS database names can be longer than 18 bytes.

  For connections to IDS V11.11 and later, database names can be up to 128 bytes.
- IDS ISAM error reporting is enabled.

  For connections to IDS V11.10 and later, ISAM errors are reported as SQLException objects, so SQLException methods can be used to obtain the error code and the message description. In addition, SQLException.printStackTrace calls display information about the cause of the ISAM errors.
- More functions are supported for connections to IDS.

  For connections to IDS 11.50 and later, the following functions are supported:
  - Progressive streaming
  - Multi-row insert operations
  - SSL support
  - Setting and Retrieving of client info properties
- Client reroute support for connections to IDS is added.

  This support requires the existence of one or more Connection Managers, a primary server, and one or more alternate servers at IDS 11.50 or later.
- Workload balancing for connections to IDS is added.

  For workload balancing to IDS, JDBC and SQLJ applications connect to a Connection Manager. They set the enableSysplexWLB property to indicate that IDS workload balancing is used.

  This support requires IDS 11.50 or later.
- Support for new IDS data types is added.

  As of IDS 11.50, IDS supports the BIGINT and BIGSERIAL data types. The IBM Data Server Driver for JDBC and SQLJ lets you access columns with those data types.

  For retrieving automatically generated keys from a BIGSERIAL column, the IBM Data Server Driver for JDBC and SQLJ adds the DB2Statement.getIDSBigSerial method.
- Support for savepoints is added.

The IBM Data Server Driver for JDBC and SQLJ supports setting of savepoints for connections to IBM Informix Dynamic Server data servers.

Connections to DB2 for IBM i enhancements

For connections to DB2 for i 6.1 and later servers, the following enhancements are added:

- Client info properties
- DECFLOAT data type
- Optimistic locking
- Progressive streaming
- User ID encryption, password encryption, and new password encryption security mechanisms
- 128-byte cursor names
- Support for methods for retrieval of automatically generated keys that require support for INSERT WITHIN SELECT SQL statements

For connections to DB2 for i5/OS V5R4 and later servers, the following enhancements are added:

- eWLM Correlator support
- IBM Data Server Driver for JDBC and SQLJ distributed transaction support

For connections to DB2 UDB for iSeries V5R3 and later servers, the following enhancements are added:

- Support for the BINARY data type
- Support for the DECIMAL data type with 63 digits of precision

Progressive streaming enhancements

For connections to DB2 for z/OS or DB2 Database for Linux, UNIX, and Windows servers, the DB2Connection.setDBProgressiveStreaming method can be used to change progressive streaming behavior after a connection to a data source is established. The DB2Connection.getDBProgressiveStreaming method can be used to determine the current progressive streaming behavior.

Global trace enhancements

Global trace settings can be changed without shutting down the driver.

You can set the db2.jcc.tracePolling global configuration property to indicate that when the following trace settings in the IBM Data Server Driver for JDBC and SQLJ global configuration file are changed while a driver instance is up, the driver modifies the trace behavior:

- db2.jcc.override.traceLevel
- db2.jcc.override.traceFile
- db2.jcc.override.traceDirectory
- db2.jcc.override.traceFileAppend

ResultSet.next behavior for DB2 connections enhancements

ResultSet.next behavior for DB2 connections can be more compatible with ResultSet.next behavior for connections with other database managers.

The allowNextOnExhaustedResultSet property can be set so that ResultSet.next behavior for a connection to DB2 for z/OS or DB2 Database for Linux, UNIX, and Windows is the same as ResultSet.next behavior for applications that are connected to an Oracle or MySQL data source. When the allowNextOnExhaustedResultSet property is set to DB2BaseDataSource.YES (1), and a forward-only cursor is positioned after the last row of a result set, a call to ResultSet.next returns `false`, instead of throwing an SQLException.

Automatically generated keys enhancements

Batched INSERT statements can return automatically generated keys.

If batch execution of a PreparedStatement object returns automatically generated keys, you can call the DB2PreparedStatement.getDBGeneratedKeys method to retrieve an array of ResultSet objects that contains the automatically generated keys. If a failure occurs during execution of a statement in a batch, you can use the DBBatchUpdateException.getDBGeneratedKeys method to retrieve any automatically generated keys that were returned.

Decimal floating-point (DECFLOAT) parameters enhancements

To enable registration of stored procedure OUT parameters as DECFLOAT, the com.ibm.db2.jcc.DB2Types.DECFLOAT JDBC data type is added.

Additional properties support

In addition to previously mentioned properties, the following Connection and DataSource properties are added:

fetchSize
Specifies the default fetch size for newly created Statement objects. This value is overridden by the Statement.setFetchSize method.

sslTrustStoreLocation
Specifies the name of the Java truststore on the client that contains the server certificate for an SSL connection.

sslTrustStorePassword
Specifies the password for the Java truststore on the client that contains the server certificate for an SSL connection.

timestampPrecisionReporting
Specifies whether trailing zeroes in a timestamp value that is retrieved from a data source are truncated.

Sample program enhancements

You can use new DB2 Java sample programs as templates to create your own application programs.

Related concepts

"New DB2 sample programs have been added" on page 89

IBM Data Server Driver Package has been enhanced

IBM Data Server Driver Package has been enhanced in Version 9.7.

In Version 9.7, IBM Data Server Driver Package supports the following additional capabilities:

- The DB2 Command Line Processor Plus (CLPPlus) for dynamically creating, editing, and running SQL statements and scripts.
- Embedded SQL applications support. No precompiler or bind capabilities are provided.
- Network share installation support (Windows only). This support enables you to install the code once (on a network share) and simply registering remote client workstations to use the driver as if it is installed locally.
- Application header files for rebuilding the PHP and Ruby drivers. These header files are also available starting in Version 9.5 Fix Pack 3.
- OLE DB support. This support is also available starting in Version 9.5 Fix Pack 3.
- DB2 Interactive Call Level Interface (db2cli) support. This support is also available starting in Version 9.5 Fix Pack 4.
- DRDA® traces (db2drdat) support. This support is also available starting in Version 9.5 Fix Pack 4.

IBM Data Server Driver Package is a lightweight deployment solution that provides runtime support for applications using ODBC, CLI, .NET, OLE DB, PHP, Ruby, JDBC, or SQLJ without the need of installing Data Server Runtime Client or Data Server Client. This driver has a small footprint and is designed to be redistributed by independent software vendors (ISVs), and to be used for application distribution in mass deployment scenarios typical of large enterprises.

Related concepts

″IBM OLE DB Provider for DB2″ in Developing ADO.NET and OLE DB Applications

″PHP application development for IBM data servers″ in pureXML Guide

″IBM data server client types″ in Installing IBM Data Server Clients

″The IBM_DB Ruby driver and Rails adapter″ in Getting Started with Database Application Development

″Network installation of IBM Data Server Driver Package (Windows)″ in Installing IBM Data Server Clients

Related reference

″db2drdat - DRDA trace ″ in Command Reference

″db2cli - DB2 interactive CLI ″ in Command Reference

Trusted context support has been extended

The IBM_DB Ruby driver, IBM PHP extensions and the IBM Data Server Provider for .NET now support trusted contexts using connection string keywords.

Trusted contexts provide a way of building much faster and more secure three-tier applications. Using trusted contexts improves performance because you do not have to get a new connection when the current user ID of the connection is switched. Also, the user's identity is always preserved for auditing and security purposes.

Related concepts

″Creating a trusted connection through IBM Data Server Provider for .NET″ in Developing ADO.NET and OLE DB Applications

″IBM Ruby driver and trusted contexts″ in Developing Perl, PHP, Python, and Ruby on Rails Applications

″Trusted contexts in PHP applications (ibm_db2)″ in Developing Perl, PHP, Python, and Ruby on Rails Applications

Related tasks

″Using trusted contexts and trusted connections″ in Database Security Guide

Sysplex support is extended to IBM data server clients and non-Java data server drivers

IBM data server clients and non-Java data server drivers that have a DB2 Connect license can now access a DB2 for z/OS Sysplex directly. Licensed clients are no longer required to go through a middle-tier IBM DB2 Connect 9.7 server to use Sysplex capabilities.

The following Sysplex capabilities are now available in IBM data server clients and the non-Java data server drivers (IBM Data Server Driver Package and IBM Data Server Driver for ODBC and CLI):

Transaction-level load balancing

Prior to the introduction of this feature, client applications that required transaction-level workload balancing had to go through a IBM DB2 Connect 9.7 server. Now, support for distributing transactions among members within a DB2 data-sharing group is available in the client, so applications accessing a DB2 for z/OS Sysplex no longer have to go through a IBM DB2 Connect 9.7 server.

Automatic client reroute with seamless failover for CLI and .NET applications

When connectivity to a member within a Sysplex is lost, the automatic client reroute feature allows the client to recover from the failure by attempting to reconnect to the database through any member of the Sysplex. Prior to the introduction of this feature, when a CLI or .NET application reestablished a database connection, an error (typically, SQL30081N) was always returned to the application to indicate that the failed transaction had been rolled back. Now, CLI or .NET applications that encounter a connectivity failure on the first SQL operation in a transaction are allowed to replay the failed SQL operation as part of automatic client reroute processing. If the connection is successful, no error is reported to the application, and the transaction is not rolled back. The connectivity failure and subsequent recovery are hidden from the application.

Clients can perform failover based on a client-specified server list (known as the alternate server list) or the server list that is returned by the database server during the last connect.

Some restrictions apply to seamless failover support.

Client-side XA support available for some transaction managers

Prior to the introduction of this feature, client-side XA support for DB2 for z/OS was not available, so non-Java client applications had to go through a IBM DB2 Connect 9.7 server to obtain XA support for DB2 for z/OS. Now, XA support for DB2 for z/OS is available in IBM data server clients and non-Java data server drivers.

Sysplex workload balancing is also supported by the IBM Data Server Driver for JDBC and SQLJ.

Related concepts

″Client Sysplex support″ in DB2 Connect User's Guide

Call Level Interface (CLI) functionality has been enhanced

Version 9.7 includes new enhancements that extend the CLI functionality, and improve the performance and reliability of applications that use CLI.

CLI applications can retrieve an accurate row count prior to fetching

You can now use the CLI statement attribute SQL_ATTR_ROWCOUNT_PREFETCH to enable a CLI application to retrieve a full row count prior to fetching.

Restriction: This feature is not supported if the cursor contains LOBs or XML data.

Prior to the introduction of this feature, calling the SQLRowCount function on a non-scrollable SELECT-only cursor set the contents of RowCountPtr to -1 because the number of rows was not available until all of the data had been fetched.

This support is also available starting in Version 9.5, Fix Pack 3 (or later).

CLI dynamic packages can be bound on demand

You can use the new SQLCreatePkg API to bind arbitrary packages to a database. This API allows you to control some BIND options.

This support is also available starting in Version 9.5, Fix Pack 3 (or later).

CLI ping capabilities have been enhanced

CLI applications can now override the default packet size used to ping a database and specify the number of times to ping the database before yielding a final result.

Prior to the introduction of this feature, the packet size was fixed, and you could ping a database one time only. These limitations made it more difficult for you to understand network complexities and judge system performance. With the new enhancements, you have finer control over the ping operation and can view accurate, more meaningful results.

This enhancement introduces two new CLI connection attributes:

SQL_ATTR_PING_REQUEST_PACKET_SIZE
Specifies the size of the ping packet that a CLI application uses when it pings a database

SQL_ATTR_PING_NTIMES
Specifies the number of times that a CLI application pings a database before a final result is returned

The application must call the SQLSetConnectAttr function to set these attributes on a connection handle before pinging a database. If you specify a value greater than 1 for SQL_ATTR_PING_NTIMES, DB2 CLI returns the average time that it took to ping the database over all iterations.

To get the current values for the new attributes, call the SQLGetConnectAttr function, passing SQL_ATTR_PING_REQUEST_PACKET_SIZE, SQL_ATTR_PING_NTIMES as arguments.

This support is also available starting in Version 9.5, Fix Pack 3 (or later).

CLI applications can improve performance when using LOAD API by specifying ANYORDER file type modifier

You can improve CLI LOAD performance by using the new statement attribute SQL_ATTR_LOAD_MODIFIED_BY to specify the ANYORDER file type modifier when using the LOAD API. The statement attribute allows you to specify multiple file type modifiers separated by spaces.

For example, the following call specifies the ANYORDER file type modifier for the CLI LOAD:

```
char *filemod="anyorder";
SQLSetStmtAttr (hstmt, SQL_ATTR_LOAD_MODIFIED_BY,
                (SQLPOINTER) filemod, SQL_NTS);
```

CLI applications can suppress header information in traces

You can suppress header information that typically appears in a CLI trace by setting the new environment attribute SQL_ATTR_TRACENOHEADER to 1. When you specify 1 for this attribute, no header information is written to the CLI trace log file. The default value for this attribute is 0.

If you accept the default or specify 0, information that is similar to the following appears in the CLI trace log file for every thread that is started:

```
[ Process: 1856, Thread: -1229691200 ]
[ Date & Time: 07/03/2008 14:43:53.074965 ]
[ Product: QDB2/LINUX DB2 v9.1.0.4 ]
[ Level Identifier: 01050107 ]
[ CLI Driver Version: 09.01.0000 ]
[ Informational Tokens: "DB2 v9.1.0.4","s080122","MI00228","Fixpack4" ]
[ Install Path: /opt/IBM/db2/V9.1.0.4 ]
[ db2cli.ini Location: /xxx/ramdisk2/db/cli/db2cli.ini ]
```

CLI applications can enable and disable the statement concentrator

You can control whether dynamic statements that contain literal values use the statement cache by setting the new **StmtConcentrator** CLI/ODBC configuration keyword or the new statement attribute SQL_ATTR_STMT_CONCENTRATOR.

By default, CLI applications use the behavior that is specified on the server.

CLI applications accessing DB2 for z/OS can roll back a transaction during streaming

CLI applications accessing DB2 for z/OS can now roll back a transaction even while in SQL_NEED_DATA state by setting the new statement attribute SQL_ATTR_FORCE_ROLLBACK. This behavior is supported when the **StreamPutData** CLI/ODBC configuration keyword set to 1.

Prior to this enhancement, CLI applications running on DB2 for z/OS had to drop and reestablish the database connection to come out of SQL_NEED_DATA state.

CLI applications can retrieve data in an interleaved fashion for LOB objects in the same row

When querying data servers that support Dynamic Data Format, CLI applications can now call SQLGetData() for previously accessed LOB columns and maintain the data offset position from the previous call to SQLGetData(). You control this behavior by specifying the new **AllowInterleavedGetData** CLI/ODBC configuration keyword or the new statement attribute SQL_ATTR_ALLOW_INTERLEAVED_GETDATA.

Prior to this enhancement, CLI applications could call SQLGetData() for previously accessed LOB columns by specifying the **AllowGetDataLOBReaccess** CLI/ODBC configuration keyword. However, there was no way to maintain the data position and offset information.

CLI applications support named parameter markers

CLI applications can now process SQL statements that contain named parameter markers represented by a colon (:) followed by a name. For example, the following syntaxes are now both supported:

```
CALL addEmp(?,?,?,?);
CALL addEmp(:empNo, :empName, :empDeptNo, :empAddr);
```

Prior to this enhancement, users were unable to pass procedure arguments in an order that differed from the order in which the parameters were defined when the procedure was created.

DB2 CLI provides no support for the ability to bind by name. DB2 CLI processes anything that matches a valid parameter marker, and treats it as if it is a normal parameter marker represented by a question mark (?).

To enable CLI support for named parameter processing, set the new **EnableNamedParameterSupport** CLI/ODBC configuration keyword to TRUE. By default named parameter processing is disabled in the driver for all servers

CLI applications support default parameters

With the new support available in Version 9.7 for default parameters in procedure definitions, you do not have to specify all parameters in the CALL statement. Unspecified parameters take the default values that are specified in the stored procedure definition.

For example, the following statement creates a procedure that has default parameter values:

```
CREATE PROCEDURE addEmp (
IN empNo       INTEGER       DEFAULT 100,
IN empName     VARCHAR(20)   DEFAULT 'nothing',
IN empDeptNo   INTEGER       DEFAULT 2,
IN empAddr     VARCHAR(100)  DEFAULT 'San Jose, CA'
) ...
```

When you call this procedure in a CLI application, you can omit specifying a value for any parameter that has a default value. The value for the missing parameter is supplied by the server. Therefore, the following example no longer results in an error:

```
CALL addEmp (empName => 'John',
empDeptNo => 1,
empAddr => 'Bangalore')
```

CLI applications support compiled compound SQL statements

You can now use compound SQL statements that include declare, begin, and end blocks in CLI applications. The statements are sent to the server as a single compound statement block. For example, the following statement is sent to the server as a single statement block:

```
BEGIN
  INSERT INTO T0 VALUES (V0);
  INSERT INTO T1 VALUES (V1);
END
```

Compound SQL statements are not supported if CLI array input chaining is used.

CLI applications can control the behavior of cursor stability scans

You can now use the **ConcurrentAccessResolution** CLI/ODBC configuration keyword to specify a prepare attribute that overrides the behavior specified for cursor (CS) stability scans. You can choose to use currently committed semantics, wait for the outcome of the transaction, or skip locked data. This setting overrides the default behavior for currently committed semantics that is defined by the **cur_commit** configuration parameter.

CLI applications support additional data type conversions and variable-length timestamp

CLI applications now support conversions between the following data types:

Table 9. Support for additional data type conversions in DB2 CLI

SQL data type	C data type
SQL_BIGINT SQL_DECIMAL SQL_DECFLOAT SQL_DOUBLE SQL_FLOAT SQL_INTEGER SQL_NUMERIC SQL_REAL SQL_SMALLINT	SQL_C_DBCHAR
SQL_TYPE_DATE	SQL_C_TYPE_TIMESTAMP SQL_C_CHAR
SQL_TYPE_TIME	SQL_C_TIMESTAMP
SQL_TYPE_TIMESTAMP	SQL_C_CHAR

In addition, CLI performs the conversions required to support a variable length timestamp of the form TIMESTAMP(p), where the precision of the timestamp value, *p*, is between 0 and 12. CLI generates truncation warnings and errors as necessary during the conversion.

The new CLI statement attribute SQL_ATTR_REPORT_TIMESTAMP_TRUNC_AS_WARN is also available to control whether a datetime overflow results in an error (SQLSTATE 22008) or warning (SQLSTATE 01S07).

Related reference

″Statement attributes (CLI) list″ in Call Level Interface Guide and Reference, Volume 2

″Connection attributes (CLI) list″ in Call Level Interface Guide and Reference, Volume 2

″CLI/ODBC configuration keywords listing by category″ in Call Level Interface Guide and Reference, Volume 2

IBM Data Server Provider for .NET has been enhanced

The IBM Data Server Provider for .NET now supports the ARRAY data type, variable length timestamps, host variables, and compound statements. Support for IBM Informix Dynamic Server has also been improved.

ARRAY data type support

Array data type support has been added to the data provider. You can use the Array data type with your stored procedure parameters. The array can be bound to a parameter in your procedure as a single argument. This simplifies the code around your SQL statements.

Compound statements support

Compound statements are supported by the provider. Using compound statements in your SQL statements can improve performance by having the statements leverage the same access plan for a group of statements.

Host variables support

Host variable support has been added to the data provider to improve compatability with other data server applications. You can use host variables (:param) in place of positioned or named parameter markers (@param). However, only one type of parameter can be used in a given statement at a time.

Variable length timestamps support

The data provider supports variable length timestamps. This makes it easier to work with other data servers. Previously the timestamp data type had a fixed 6–digit precision. The timestamp data type has been improved to support 0 to 12 digits of precision.

Other data server support enhancements

The provider works with multiple types of IBM data servers. Version 9.7 includes enhancements that improve the performance of .NET applications that connect to IBM Informix Dynamic Server data servers.

ReturnValue parameter support for stored procedures

Informix data server stored procedures are able to return single or multiple result sets. Previously, the data provider did not support multiple values from user defined routines (UDRs). The added support for ReturnValue parameters means that the data provider is able to retrieve the result set as a single return value.

BIGINT and BIGSERIAL data types support

The provider previously supported only INT8 and SERIAL8 data types for

64–bit integers. Support has been added for BIGINT and BIGSERIAL for Informix data servers which have better performance compared to the previous two data types.

IBM Informix Dynamic Server: High availability disaster recovery (HADR) support

High availability disaster recovery is a data replication feature that protects against data loss by replicating data to a secondary database. The IBM Data Server Provider for .NET will work with this feature if you have it setup on your data servers.

IBM Informix Dynamic Server: Workload Manager (WLM) support

The Workload Manager is a feature that you can use to maximize the utilization of your resources to means your specific goals. The IBM Data Server Provider for .NET will work with this feature if you have it setup on your data servers.

DB2 for z/OS: Seamless failover in XA support

Seamless failover is a feature that improves XA connection reliability for data server applications. The IBM Data Server Provider for .NET will work with this feature if you have it setup on your data servers.

Related concepts

″IBM Data Server Provider for .NET″ in Developing ADO.NET and OLE DB Applications

Related tasks

″Deploying .NET applications (Windows)″ in Developing ADO.NET and OLE DB Applications

Chapter 12. SQL Procedural Language (SQL PL) enhancements

Version 9.7 contains numerous improvements that make it easier to work with SQL Procedural Language (SQL PL).

The following SQL PL enhancements are included:

- Related database objects can be grouped in named sets (modules) and can be reused (see "Related database object definitions can be encapsulated in new module database object")
- Compiled compound statements support (see "Compiled compound statement support has been added" on page 108)
- User-defined functions support (see "SQL PL functionality has been extended for user-defined functions" on page 109)
- Triggers support (see "Triggers support has been extended" on page 109)
- New data types support in SQL PL applications: anchored data type, Boolean data type, associative array data type, cursor data type, and row data type (see "New data types are supported" on page 111)

You can also use system-defined modules for performing a variety of application development tasks in SQL PL. For more information, see "System-defined modules simplify SQL PL and application logic" on page 86.

Related database object definitions can be encapsulated in new module database object

New module database objects simplify database design and application development by allowing you to group together, in a named set within a schema, a collection of related data type definitions, database object definitions, routine prototypes, routines and other logic elements.

This simple encapsulation of related database elements also facilitates easy deployment of the definitions to other schemas or databases.

You can perform the following actions using modules:

- Define within a single object definition the related definitions for any of:
  - SQL procedures
  - SQL functions
  - External procedures
  - External functions
  - Global conditions
  - A module initialization procedure for implicit execution upon module initialization
  - User-defined data type definitions including: distinct type, array type, associative array type, row type, and cursor type
  - Global variables
- Define a namespace such that objects defined within the module can refer to other objects defined in the module without providing an explicit qualifier.

- Add object definitions that are private to the module. These objects can only be referenced by other objects within the module.
- Add object definitions that are published. Published objects can be referenced from within the module or from outside of the module.
- Define published prototypes of routines without routine bodies in modules and add the routines with routine bodies later, using the same signature as the routine prototype.
- Define a module initialization procedure that is automatically executed when the first reference is made to a module routine or module global variable. This procedure can include SQL statements, SQL PL statements, and can be used to set default values for global variables or to open cursors.
- Reference objects defined in the module from within the module and from outside of the module by using the module name as a qualifier (2-part name support) or a combination of the module name and schema name as qualifiers (3-part name support).
- Drop objects defined within the module.
- Drop the module.
- Manage who can reference objects in a module by allowing you to grant and revoke the EXECUTE privilege for the module.
- Port similar database objects written in other procedural languages to DB2 SQL PL.

Modules can be created using the CREATE MODULE statement.

You can use the modules.db2 sample program to learn how to use this feature.

Related concepts

“New DB2 sample programs have been added” on page 89

″Modules″ in SQL Procedural Languages: Application Enablement and Support

″Referencing objects defined in modules″ in SQL Procedural Languages: Application Enablement and Support

Related tasks

″Creating modules″ in SQL Procedural Languages: Application Enablement and Support

″Dropping modules″ in SQL Procedural Languages: Application Enablement and Support

Related reference

″CREATE MODULE ″ in SQL Reference, Volume 2

Compiled compound statement support has been added

Starting in Version 9.7, new compiled compound statements extend existing compound statement support by supporting more SQL PL statements and SQL language elements.

A compound statement is a BEGIN-END block that includes SQL and procedural statements. This statement is similar to an inline compound statement (previously called a SQL dynamic compound statement) except that it can contain many more SQL PL statements and language elements. A compiled compound statement provides support similar to what is supported in an SQL procedure body, but with some restrictions. Compiled compound statements can be executed within applications or interactively from the DB2 Command Line Processor, CLPPlus processor, and other supported DB2 interfaces.

The extended support for compound SQL resulted in renaming the following statements in the documentation:

- compound SQL (compiled) replaces compound SQL (procedure)
- compound SQL (inlined) replaces compound SQL (dynamic)

Related reference

″Compound SQL (compiled) ″ in SQL Reference, Volume 2

Triggers support has been extended

In Version 9.7, an enhanced set of SQL PL features can be referenced in triggers when triggers are created with a compiled compound statement as the trigger body.

In previous releases, triggers could only contain the subset of SQL PL statements known as inline SQL PL statements. In Version 9.7, triggers can be defined using a compiled compound statement that can include or reference the following features:

- SQL PL statements, including CASE and REPEAT statements
- Support for declaring and referencing variables defined by local user-defined data types including: row data types, array data types and cursor data types
- Cursor declarations
- Dynamic SQL
- Conditions
- Condition handlers

Related concepts

″Triggers (PL/SQL)″ in SQL Procedural Languages: Application Enablement and Support

Related reference

″CREATE TRIGGER statement (PL/SQL)″ in SQL Procedural Languages: Application Enablement and Support

SQL PL functionality has been extended for user-defined functions

In Version 9.7, an enhanced set of SQL PL features can be referenced in SQL functions when functions are created with a compiled compound statement as the function-body.

In previous releases, SQL functions could only contain the subset of SQL PL statements known as inline SQL PL statements.

In Version 9.7, SQL functions can be defined using a compiled compound statement that include or reference the following features:

- SQL PL statements, including CASE and REPEAT statements
- Support for declaring and referencing variables defined by local user-defined data types including: row data types, array data types and cursor data types
- Cursor declarations
- Dynamic SQL
- Conditions
- Condition handlers

In releases prior to Version 9.7, these features are either not available or are only available for use within SQL procedures.

Examples

The following example demonstrates the difference required in a CREATE FUNCTION statement to create a compiled SQL function instead of an inline SQL function.

Table 10. Comparison of SQL syntax required for inline SQL functions and compiled SQL functions

Inline SQL function definition	Compiled SQL function definition

`CREATE FUNCTION TAN (X DOUBLE)`
`RETURNS DOUBLE`
`LANGUAGE SQL`
`CONTAINS SQL`
`NO EXTERNAL ACTION`
`DETERMINISTIC`
`BEGIN` **`ATOMIC`**
`  RETURN SIN(X)/COS(X);`
`END`
	`CREATE FUNCTION TAN (X DOUBLE)`
	`RETURNS DOUBLE`
	`LANGUAGE SQL`
	`CONTAINS SQL`
	`NO EXTERNAL ACTION`
	`DETERMINISTIC`
	`BEGIN`
	`  RETURN SIN(X)/COS(X);`
	`END`

The following example demonstrates a compiled SQL function definition containing a cursor, condition handler statement, and a REPEAT statement:

```
CREATE FUNCTION exit_func( a integer)
SPECIFIC udfPSM320
LANGUAGE SQL
RETURNS integer
BEGIN
  DECLARE val INTEGER DEFAULT 0;

  DECLARE myint INTEGER DEFAULT 0;

  DECLARE cur2 CURSOR FOR
   SELECT c2 FROM udfd1
   WHERE c1 <= a
  ORDER BY c1;

  DECLARE EXIT HANDLER FOR NOT FOUND
  BEGIN
    SIGNAL SQLSTATE '70001'
     SET MESSAGE_TEXT =
      'Exit handler for not found fired';
  END;

OPEN cur2;

REPEAT
  FETCH cur2 INTO val;
  SET myint = myint + val;
UNTIL (myint >= a)
END REPEAT;

CLOSE cur2;

RETURN myint;

END@
DB20000I  The SQL command completed
 successfully.
```

The compiled SQL function can be invoked by executing the following SQL statement:

```
VALUES(exit_func(-1));
```

The output of this invocation, which demonstrates the successful firing of the exit handler, is as follows:

```
1
-----------
SQL0438N  Application raised error or warning with
diagnostic text: "Exit handler for not found fired".
SQLSTATE=70001
```

Related reference

″SECOND ″ in SQL Reference, Volume 1

″FETCH ″ in SQL Reference, Volume 2

″INSERT ″ in SQL Reference, Volume 2

″SELECT INTO ″ in SQL Reference, Volume 2

″SET variable ″ in SQL Reference, Volume 2

New data types are supported

You can use new data types to greatly simplify SQL PL logic.

You can use these data types to facilitate the DB2 enablement of applications written in other procedural SQL languages that support a similar data type. These data types can be used in the following contexts:

- Compound SQL (compiled) statements.
- Parameter types in SQL procedures.
- Parameter types in SQL functions whose function-body is a compound SQL (compiled) statement.
- Return types in SQL functions whose function-body is a compound SQL (compiled) statement.
- Global variables.
- User-defined type definitions for array, cursor, or row types. Anchored types can also be used in these type definitions and in distinct type definitions.

Anchored data type has been added

You can use a new anchored data type for use within SQL Procedural Language (SQL PL) applications. The anchored data type is used to assign a data type that is, and will always remain the same type, as that of another object.

This support is useful when it is necessary for a variable to maintain the same data type as another object if they have a logical relationship or when the data type is not yet known.

This data type can also be used to hold the values of a given column or row in a table to enforce and maintain data type compatibility. If a column data type is changed or the column definitions of a table are changed, a corresponding change might be required to a parameter or variable in a PL/SQL block. Instead of coding the specific data type into the variable declaration, an anchor data type declaration can be used instead.

Related concepts

″Anchored data type″ in SQL Procedural Languages: Application Enablement and Support

″Anchored data type variables″ in SQL Procedural Languages: Application Enablement and Support

″Restrictions on the anchored data type″ in SQL Procedural Languages: Application Enablement and Support

″Features of the anchored data type″ in SQL Procedural Languages: Application Enablement and Support

″Examples: Anchored data type use″ in SQL Procedural Languages: Application Enablement and Support

Related tasks

″Declaring local variables of the anchored data type″ in SQL Procedural Languages: Application Enablement and Support

Related reference

″Anchored types″ in SQL Reference, Volume 1

Boolean data type has been added

You can use a new system-defined Boolean data type for use within SQL Procedural Language (SQL PL) applications that provides support for declaring and referencing the system-defined logical values: TRUE, FALSE, or NULL within compound SQL (compiled) statements.

The Boolean data type is like any other built-in type, such that it can also be referenced in expressions and assigned the resulting value of a logical expression.

Example

The following is an example of the creation of a Boolean variable and the setting of it to the value TRUE:

```
CREATE VARIABLE gb BOOLEAN;
SET gb = TRUE;
```

The following is an example of a simple SQL function that accepts a Boolean parameter value and also returns a Boolean value:

```
CREATE FUNCTION fb1(p1 BOOLEAN, p2 INT) RETURNS BOOLEAN
BEGIN
  IF p1 = TRUE AND p2=1 THEN
    RETURN p1;
  ELSE
    RETURN FALSE;
  END IF;
END
```

The following is an example of how to set the variable with the output function fb1:

```
SET gb = fb1(TRUE,1);
```

Related concepts

″Boolean data type″ in SQL Procedural Languages: Application Enablement and Support

Related reference

″Boolean values″ in SQL Reference, Volume 1

Associative array data type has been added

You can use a new user-defined associative array data type for use within SQL Procedural Language (SQL PL) applications. You can use it to simplify the manipulation of data within your application because you manage and pass sets of values of the same kind in the form of a collection.

Associative arrays provide the following features:

- Because the array has no predefined cardinality, you can continue adding elements to the array without concern for a maximum size which is useful if you do not know in advance how many elements will constitute a set.
- The array index value can be a non-integer data type. VARCHAR and INTEGER are the supported index data types for the associative array index.
- Array index values are unique, are of the same data type, and do not have to be contiguous. In contrast to a conventional array which is indexed by position, an associative array is an array that is indexed by values of another data type and there are not necessarily index elements for all possible index values between the lowest and highest. This is useful if, for example, you want to create a set that stores names and phone numbers. Pairs of data values can be added to the set in any order are stored according to the order of array index values.
- Array data can be accessed and set using direct references or by using a set of available array functions. For the list of array functions, see the "Supported functions and administrative SQL routines and views" topic.

Related concepts

″Associative array data type″ in SQL Procedural Languages: Application Enablement and Support

Related reference

″Supported functions and administrative SQL routines and views″ in SQL Reference, Volume 1

″CREATE TYPE (array) ″ in SQL Reference, Volume 2

″Array values″ in SQL Reference, Volume 1

Cursor data type support has been added

You can use the built-in CURSOR data type or implement a user-defined cursor data type for use within SQL Procedural Language (SQL PL) applications, making it easier to work with result set data.

This support allows you to define a cursor data type after which parameters and variables of the defined cursor type can be declared. Cursor parameters and variables are like programmatic updatable pointers to a cursor; they hold a reference to the context of a cursor. Previously, cursors could only be used to hold a single predefined constant result set value and, as such, were comparable to a static constant programming value. This new support enables you to pass cursors between routines and to work with cursor data when the SQL statement that defines the cursor is not known or might change.

Variables or parameters of a cursor type can be:

- Uninitialized at creation time
- Assigned a result set definition based on an SQL statement
- Set to another result set definition
- Used as a procedure output parameter
- Specified as parameters to SQL procedures or SQL functions
- Specified as the return value from SQL functions

A cursor value can include the specification of parameters used in the associated query. This is known as a parameterized cursor. When a parameterized cursor is opened, argument values are provided for the defined parameters that are used in the query. This allows an OPEN using a cursor variable to provide input values similar to the use of parameter markers in dynamic cursors, or the use of host variables in statically declared cursors.

Related concepts

″Cursor types″ in SQL Procedural Languages: Application Enablement and Support

″Overview of cursor data types″ in SQL Procedural Languages: Application Enablement and Support

″Example: Cursor variable use″ in SQL Procedural Languages: Application Enablement and Support

Related tasks

″Creating cursor data types using the CREATE TYPE statement″ in SQL Procedural Languages: Application Enablement and Support

Related reference

″Cursor values″ in SQL Reference, Volume 1

″CREATE TYPE (cursor) ″ in SQL Reference, Volume 2

Row data type has been added

You can use a new user-defined row data type for use within SQL Procedural Language (SQL PL) applications. This data type is a structure composed of multiple fields, each with its own name and data type, which can be used to store the column values of a row in a result set or other similarly formatted data.

You must create this user-defined data type using the CREATE TYPE statement before you can reference it.

You can use this data type for the following tasks:

- Creating or declaring of variables of type row that can be used to store row data.
- Passing row values as parameters to other SQL routines.
- Storing multiple SQL data type values as a single set. For example, database applications process records one at a time and require parameters and variables to temporarily store records. A single row data type can replace the multiple parameters and variables required to otherwise process and store the record values.
- Referencing row data in data change statements and queries, including INSERT, FETCH, and SELECT INTO.

Related concepts

″Row types″ in SQL Procedural Languages: Application Enablement and Support

″Creating row variables″ in SQL Procedural Languages: Application Enablement and Support

″Referencing row values″ in SQL Procedural Languages: Application Enablement and Support

″Comparing row variables and row field values″ in SQL Procedural Languages: Application Enablement and Support

″Passing rows as routine parameters″ in SQL Procedural Languages: Application Enablement and Support

″Examples: Row data type use″ in SQL Procedural Languages: Application Enablement and Support

″Assigning values to row variables″ in SQL Procedural Languages: Application Enablement and Support

Related reference

″CREATE TYPE (row) ″ in SQL Reference, Volume 2

Chapter 13. Net Search Extender enhancements

Version 9.7 includes enhancements that extend the Net Search Extender functionality.

Full-text searches are available in DB2 Version 9.7 in the following new scenarios:

- Partitioned tables (see “Full-text searches support partitioned tables”)
- Additional partitioned database environments (see “Full-text searches support in partitioned database environments has been extended”)

In addition, you can choose a new option that uses the results of integrity process to perform some incremental updates operations. For more information, see “Incremental update based on integrity processing is supported.”

Full-text searches support partitioned tables

Starting in Version 9.7, you can create and maintain text search indexes on partitioned tables. Any partitioning feature combination of the base table is supported.

The text search index is not partitioned according to the defined ranges. However, if the partitioned table is distributed across multiple nodes in a partitioned database environment, the index is partitioned per partition in the same way as for a table that is not a partitioned table.

Full-text searches support in partitioned database environments has been extended

In Version 9.7, you can use Net Search Extender (NSE) full-text search in all partitioned database environments except for the Linux on Power servers, Solaris x64 (Intel® 64 or AMD64), and Microsoft Cluster Server (MSCS) environments.

Prior to Version 9.7, full-text searches were supported in partitioned database environments on the AIX operating system.

Incremental update based on integrity processing is supported

You can use the new **AUXLOG** option of the CREATE INDEX Net Search Extender command to perform certain incremental update operations based on the results of integrity processing. This enables, for example, a synchronization of the text index after a bulk insert of data with the load utility.

Data synchronization in Net Search Extender is based on triggers that update a log table whenever the triggers capture information about new, changed, and deleted documents. There is one log table for each text index. Applying the information in the log table to the corresponding text index is referred to as performing an *incremental update*.

If you specify the **AUXLOG** option, information about new and deleted documents is captured through integrity processing in an auxiliary staging table maintained by Net Search Extender, and information about changed documents is captured through triggers and stored in the base log table.

This option is enabled by default for partitioned tables and is disabled for nonpartitioned tables.

Chapter 14. Installation, upgrade, and fix pack enhancements

Version 9.7 includes enhancements that make it faster to deploy products and easier to maintain them.

Response files support is improved with the following enhancements:

- db2rspgn (response file generator) command support on Linux and UNIX operating systems (see "db2rspgn command supported on Linux and UNIX operating systems" on page 120)
- Additional response file support for uninstalling DB2 products (see "Uninstalling using a response file is supported in more situations" on page 121)
- A new response file keyword, **UPGRADE_PRIOR_VERSIONS** (see "UPGRADE_PRIOR_VERSIONS response file keyword has been added" on page 121)

Deployment of DB2 product installation is enhanced with shared DB2 copy support, see "Instances and DB2 Administration Server can be created in a shared DB2 copy (Linux and UNIX)" on page 120.

Product installations on all operating systems have been improved as follows:

- New commands for validating DB2 product installations and for starting the product update service utility (see "Product installation can be validated using db2val command" on page 122 and "Product update service support has been extended" on page 122)
- Additional support for the IBM Tivoli System Automation for Multiplatforms (SA MP) component (see "IBM Tivoli System Automation for Multiplatforms (SA MP) support has been extended (AIX, Linux and Solaris)" on page 124)
- Non-root installation support for IBM Database Add-Ins for Visual Studio (see "IBM Database Add-Ins for Visual Studio can be installed by more users" on page 124)

Product installations on Linux and UNIX operating systems benefit with these operating-system specific enhancements to commands:

- db2iprune (reduce installation image size command) support (see "Installation images can be reduced (Linux and UNIX)" on page 124)
- db2updserv (show product updates command) support (see "Product update service support has been extended" on page 122)
- New commands to manually create or remove the DB2 tool entries (see "Product installation on Linux and UNIX platforms has been enhanced" on page 123)
- db2ls (list installed DB2 products and features command) support from the installation media (see "Product installation on Linux and UNIX platforms has been enhanced" on page 123)
- Updates to instance commands (see "Product installation on Linux and UNIX platforms has been enhanced" on page 123)

Product maintenance has been simplified when applying fix packs with the following enhancements:

- Additional support for universal fix packs (see "Universal fix packs support has been extended (Windows)" on page 125)

- An option to not back up some files during installation (see “Fix pack installations can require less space (Linux and UNIX)” on page 125)

If you have a Version 8 or Version 9 copy installed and you want to use Version 9.7 instead, you need to upgrade to Version 9.7. DB2 Version 9.7 is a new release. You cannot apply a fix pack to upgrade from a Version 9 copy to Version 9.7.

To learn about upgrade limitations, possible issues, and other details, see “Upgrade essentials for DB2 servers” in *Upgrading to DB2 Version 9.7* and “Upgrade essentials for clients” in *Upgrading to DB2 Version 9.7*.

Upgrading your DB2 servers and DB2 clients to Version 9.7 might require that you also upgrade your database applications and routines. To help determine whether you must upgrade, see the “Upgrade essentials for database applications ” in *Upgrading to DB2 Version 9.7* and “Upgrade essentials for routines” in *Upgrading to DB2 Version 9.7* topics.

Instances and DB2 Administration Server can be created in a shared DB2 copy (Linux and UNIX)

Starting in Version 9.7, you can create instances and a DB2 Administration Server (DAS) in a shared DB2 copy on an AIX system workload partitions, a Network File System shared server, or on Solaris Zones.

The following shared systems are supported:

AIX system workload partitions (WPARs)
A DB2 copy is installed on the global environment and shared on system WPARs with read-only permission. For AIX WPAR, fix pack updates are also supported.

Network File System (NFS) shared server
A DB2 copy is installed on NFS server and shared (normally with read-only permission) on NFS clients.

Solaris Zones
A DB2 copy is installed on global zone on Solaris and shared on other zones with read-only permission.

Related concepts

″DB2 products in a workload partition (AIX)″ in Installing DB2 Servers

Related reference

″db2icrt - Create instance ″ in Command Reference

db2rspgn command supported on Linux and UNIX operating systems

Starting with Version 9.7, you can use the response file generator command, db2rspgn, on Linux and UNIX operating systems to re-create an installation setup on other computers.

Prior to Version 9.7, the response file generator command, db2rspgn, was supported on Windows operating systems only.

The db2rspgn command automatically extracts the customized DB2 product, feature, and instance configuration profiles and saves them to response files and instance configuration profiles. You can use the generated response files and instance configuration profiles to manually re-create the exact setup on other machines.

Related concepts

″The response file generator″ in Installing DB2 Servers

Related reference

″db2rspgn - Response file generator ″ in Command Reference

Uninstalling using a response file is supported in more situations

You can now use a response file to uninstall DB2 products, features, or languages on Linux, UNIX, and Windows operating systems. On Linux and UNIX operating systems, you can also use a response file to uninstall the DB2 Information Center.

Prior to Version 9.7, you could use a response file only to uninstall a DB2 product on Windows operating systems or to uninstall a DB2 feature on Linux and UNIX operating systems.

Uninstalling using a response file includes the following benefits:

- You do not have to provide input while uninstalling.
- You can uninstall multiple products, features, or languages simultaneously.
- You can share the response file among many systems to remove the same set of products, features, or languages.

A sample response file for uninstalling, db2un.rsp, is provided on the product DVD in *image*/db2/*platform*/samples, where *platform* refers to the applicable hardware platform. During DB2 product installation, this sample response file is copied to *DB2DIR*/install, where *DB2DIR* is the path where the DB2 product was installed.

To uninstall DB2 products, features, and languages within a DB2 copy:

- On Linux and UNIX operating systems, use the db2_deinstall command with the **-r** option.
- On Windows operating systems, use the db2unins command with the **-u** option.

To uninstall the DB2 Information Center on Linux operating systems, use the doce_deinstall command with the **-r** option.

Related concepts

″Response file uninstall basics″ in Installing DB2 Servers

Related reference

″Response file keywords″ in Installing DB2 Servers

″db2unins - Uninstall DB2 database products, features, or languages ″ in Command Reference

″db2_deinstall - Uninstall DB2 products, features, or languages ″ in Command Reference

UPGRADE_PRIOR_VERSIONS response file keyword has been added

You can use the new **UPGRADE_PRIOR_VERSIONS** response file keyword to specify the version of a DB2 product to be upgraded. The new keyword is supported on Linux, UNIX, and Windows operating systems. However, on Linux and UNIX operating systems, this keyword is valid only for non-root upgrades.

This keyword replaces the **MIGRATE_PRIOR_VERSIONS** keyword, which has been deprecated.

A response file is an ASCII text file that contains setup and configuration information. Unlike using the DB2 Setup wizard to install, deploy, or uninstall products, features, or languages, using a response file lets you perform these tasks without interaction. Ready-to-use sample response files with default entries are included on the DB2 DVD. The sample response files are located in db2/*platform*/samples, where *platform* refers to the hardware platform.

Related concepts

"Some response file keywords have been deprecated" on page 185

"INTERACTIVE response file keyword has been changed" on page 144

Related reference

″Response file keywords″ in Installing DB2 Servers

Product installation can be validated using db2val command

The new db2val tool verifies the core functionality of a DB2 copy by validating the installation, instances, database creation, connections to the database and health of partitioned database environments.

This validation can be helpful if you manually deployed a DB2 copy on Linux and UNIX operating systems by using tar.gz files. The db2val command can quickly ensure that you correctly configured the copy and that the copy is what you expect it to be.

Related tasks

″Validating your DB2 copy″ in Installing DB2 Servers

Related reference

″db2val - DB2 copy validation tool command″ in Command Reference

Product update service support has been extended

Support for the product update service has been extended in Version 9.7 to include the Linux and UNIX platforms. You can also use the new db2updserv command to start the product update service from a command prompt.

The update service allows you to stay informed of product updates such as:

- Messages about releases and updates of the DB2 products.
- Availability of technical materials such as tutorials, web cast, and white papers.
- IBM Marketing activities pertinent to your area of interest.

The update service is enabled by default during DB2 product installations. You can access product updates at any time using any of the following ways:

- Using the new db2updserv command
- Using First Steps
- Using the shortcuts in the Start menu.

To use update service, ensure the update service component is installed. This can be done by choosing a typical install (if using a response file installation, set INSTALL_TYPE = TYPICAL) or a custom install with the DB2 Update Service component selected (if using a response file installation, set INSTALL_TYPE = CUSTOM and COMP = DB2_UPDATE_SERVICE).

Related concepts

″First Steps interface″ in Installing DB2 Servers

Related tasks

″Checking for DB2 updates″ in Installing DB2 Servers

Related reference

″db2updserv - Show product updates″ in Command Reference

Product installation on Linux and UNIX platforms has been enhanced

On Linux and UNIX operating systems, new capabilities have been added to simplify the product installation and the management of DB2 instances.

Version 9.7 includes the following enhancements:

- Instance activities are logged when you perform the following tasks:
  - Create an instance by using the db2icrt and db2nrcfg commands
  - Drop an instance by using the db2idrop command
  - Update an instance by using the db2iupdt and db2nrupdt commands
  - Upgrade an instance by using the db2iupgrade and db2nrupgrade commands

  A log file, sqllib/log/db2instance.log, is created during instance creation to record the instance activities. This file is deleted if you drop the instance.
- Old instance entries in the DB2 global registry are cleaned up on all nodes when you issue the db2icrt, db2idrop, db2iupgrade, or db2iupdt command in partitioned database environments. This cleanup is not performed on any pre-DB2 Version 9.7 instances after upgrading.
- You can now run the db2ls command from the installation media. This command lists the installed DB2 products and features.
- On Linux operating systems, for the DB2 tools already installed, you can now add the following tools to the Main menu:
  - Check for DB2 Updates
  - Command Line Processor
  - Command Line Processor Plus
  - Configuration Assistant
  - Control Center
  - First Steps
  - Query Patroller.

  The following new commands can be run to manually create or remove the DB2 tool entries:
  - db2addicons
  - db2rmicons

Related concepts

″Main menu entries for DB2 tools (Linux)″ in Installing DB2 Servers

Related tasks

″Listing DB2 database products installed on your system (Linux and UNIX)″ in Installing DB2 Servers

IBM Tivoli System Automation for Multiplatforms (SA MP) support has been extended (AIX, Linux and Solaris)

IBM Tivoli SA MP support has been extended in Version 9.7 to include Solaris SPARC.

IBM Tivoli SA MP, Version 3.1 Fix Pack 1, is now bundled with the IBM Data Server on Solaris SPARC, Linux and AIX operating systems.

SA MP Version 3.1 Fix Pack 1 is integrated with DB2 products on AIX, Linux, and Solaris SPARC 10. SA MP Version 3.1 Fix Pack 1 is not supported on AIX system workload partitions (WPARs), Solaris 9, Solaris 10 non-global zones, or Solaris AMD64.

Related tasks

″Installing and upgrading the SA MP with the DB2 installer″ in Installing DB2 Servers

Related reference

″Supported software and hardware for IBMTivoli System Automation for Multiplatforms (SA MP)″ in Installing DB2 Servers

IBM Database Add-Ins for Visual Studio can be installed by more users

You can now install IBM Database Add-Ins for Visual Studio with a non-administrator account with elevated privileges.

IBM Database Add-Ins for Visual Studio provides tools for rapid application development, database schema development, and debugging.

Related concepts

″DB2 integration in Visual Studio″ in Developing ADO.NET and OLE DB Applications

Installation images can be reduced (Linux and UNIX)

In Version 9.7, you can use the db2iprune command on the Linux and UNIX operating systems.

Prior to Version 9.7, you could use this command to reduce the size of DB2 product installation images and fix pack images on Windows operating systems only.

This tool is useful for large-scale DB2 deployments and for embedding DB2 products within an application. The db2iprune command removes files associated with unwanted products, features, and languages based on an input file. The result is a smaller DB2 installation image that can be installed using the regular DB2 installation methods

Related tasks

″Reducing the size of your DB2 fix pack installation image″ in Installing DB2 Servers

Related reference

″db2iprune - Reduce installation image size command″ in Command Reference

Universal fix packs support has been extended (Windows)

Starting with Version 9.7, you have two choices on the Windows operating systems for installing a fix pack: a universal fix pack, which applies to all products, or a product-specific fix pack.

You can use a universal fix pack to service multiple DB2 products installed in an installation path. To upgrade a single product or to install a product in a new path, use a product-specific fix pack.

You do not need a universal fix pack if the installed DB2 products are only DB2 server products or a Data Server Client. In this case, use the single server image fix pack.

Universal fix packs were already available on Linux and UNIX platforms.

Related tasks

″Applying fix packs″ in Installing and Configuring DB2 Connect Servers

Fix pack installations can require less space (Linux and UNIX)

On Linux and UNIX operating systems, you can reduce the amount of space required for a fix pack installation by using the installFixPack command with the new **-f nobackup** parameter.

If you specify the **-f nobackup** parameter, the installation files are not backed up when the components are updated, which saves space.

Related reference

″installFixPack - Update installed DB2 products ″ in Command Reference

Chapter 15. DB2 Connect enhancements and changes summary

In Version 9.7, DB2 Connect capabilities have been enhanced and changed.

DB2 Connect overview

DB2 Connect provides fast and robust connectivity to IBM mainframe databases for e-business and other applications running on Linux, UNIX, and Windows operating systems.

DB2 for i, DB2 for z/OS, and DB2 Server for VM and VSE continue to be the database systems of choice for managing critical data for the largest organizations in the world. There is a great demand to integrate this data with applications running on Linux, UNIX, and Windows operating systems.

DB2 Connect has several connection solutions, including DB2 Connect Personal Edition and a number of DB2 Connect server products. A DB2 Connect server is a server that concentrates and manages connections from multiple desktop clients and Web applications to DB2 database servers running on mainframes or IBM Power Systems® servers.

Version 9.7 enhancements and changes

The following Version 9.7 enhancements and changes affect the functionality and capabilities of DB2 Connect.

Product packaging enhancements

- "Component names have changed" on page 3

Security enhancements

- "SSL client support expanded and configuration simplified" on page 66
- "AES encryption of user ID and password enhances security" on page 66

Application development enhancements

- "Common SQL API supports the development of portable administrative applications" on page 87
- "Python application development support has been added" on page 85
- "IBM Database Add-Ins for Visual Studio have been enhanced" on page 88

IBM data server client and driver enhancements

- "JDBC and SQLJ support has been enhanced" on page 92
- " IBM Data Server Driver Package has been enhanced" on page 98
- "IBM Data Server Provider for .NET has been enhanced" on page 105
- "Trusted context support has been extended" on page 99
- "Sysplex support is extended to IBM data server clients and non-Java data server drivers" on page 100
- "Call Level Interface (CLI) functionality has been enhanced" on page 101

High availability, backup, logging, resiliency, and recovery enhancements

- "Enhanced resilience to errors and traps reduces outages" on page 37
- "Administration notification and diagnostic logs occupy specified amount of disk space" on page 38

Installation, upgrade, and fix pack enhancements

- "Instances and DB2 Administration Server can be created in a shared DB2 copy (Linux and UNIX)" on page 120
- "Installation images can be reduced (Linux and UNIX)" on page 124
- "Product installation can be validated using db2val command" on page 122
- "Product update service support has been extended" on page 122
- "Product installation on Linux and UNIX platforms has been enhanced" on page 123
- "IBM Database Add-Ins for Visual Studio can be installed by more users" on page 124
- "Universal fix packs support has been extended (Windows)" on page 125
- "Fix pack installations can require less space (Linux and UNIX)" on page 125
- "Uninstalling using a response file is supported in more situations" on page 121
- "UPGRADE_PRIOR_VERSIONS response file keyword has been added" on page 121
- "db2rspgn command supported on Linux and UNIX operating systems" on page 120

Administration changes

- "Some database manager configuration parameters have been changed" on page 133
- "Some registry and environment variables have changed" on page 136
- "DESCRIBE command lists information about additional index types" on page 140
- "Registry files have been removed from the DB2 installation path" on page 144

Security changes

- "System administrator (SYSADM) authority scope has changed" on page 145
- "Security administrator (SECADM) abilities have been extended" on page 147
- "Database administrator (DBADM) authority scope has changed" on page 148
- "SSLconfig.ini and SSLClientconfig.ini files replaced with new database manager configuration parameters" on page 149

Application development changes

- "Merge modules for ODBC, CLI, and .NET have been combined (Windows)" on page 169

Deprecated functionality

- "db2iupdt command -s option has been deprecated (Linux and UNIX)" on page 183
- "Control Center tools and DB2 administration server (DAS) have been deprecated" on page 174
- "Health monitor has been deprecated" on page 177

- “Some response file keywords have been deprecated” on page 185
- “Instance and database migration commands and APIs have been deprecated” on page 184
- “Worksheet Format (WSF) for Export and Load utilities has been deprecated” on page 178

Discontinued functionality

- “db2secv82 command has been discontinued” on page 190
- “db2ilist command options -a and -p have been discontinued” on page 192
- “Netscape browser support has been discontinued” on page 188

Part 2. What's changed

DB2 Version 9.7 for Linux, UNIX, and Windows contains changed functionality, deprecated functionality, and discontinued functionality that you should keep in mind when coding new applications or when modifying existing applications.

Being aware of these changes facilitates your current application development and plans to upgrade to DB2 Version 9.7.

Chapter 16, "Changed functionality," on page 133
This chapter describes the changes to existing DB2 functionality, including changes related to database setup, database administration, application development, and CLP and system commands.

Chapter 17, "Deprecated functionality," on page 173
This chapter lists the deprecated functionality, which refers to specific functions or features that are supported but are no longer recommended and might be removed in a future release.

Chapter 18, "Discontinued functionality," on page 187
This chapter lists features and functionality that are unsupported in Version 9.7.

Chapter 19, "Version 9 deprecated and discontinued DB2 functionality summary," on page 193
This chapter lists features and functionality that have been deprecated or discontinued in DB2 Version 9.1, Version 9.5, and Version 9.7.

In Version 9.7, IBM has updated the list of DB2 database products and features. To read about these products changes and to view the related licensing and marketing information, see the DB2 Version 9 for Linux, UNIX, and Windows home page at http://www.ibm.com/db2/9.

Chapter 16. Changed functionality

Changed functionality typically involves changes in default values or an outcome different from what would have occurred in previous releases. For example, an SQL statement that you used in Version 9.5 might produce different results in Version 9.7.

Administration changes summary

Version 9.7 includes changed functionality that affects how you administer and work with DB2 databases.

Partitioned indexes are created by default for partitioned tables

Starting in Version 9.7, if you do not specify the PARTITIONED or the NOT PARTITIONED clause on the CREATE INDEX statement when creating indexes on partitioned tables, a partitioned index is created by default.

Details

When you create an index for a partitioned table, by default, the index is created as a partitioned index, unless the following situations apply:

- You specify UNIQUE on the CREATE INDEX statement and the index key does not include all the table-partitioning key columns.
- You create an index over spatial data.
- You create an index over XML data.

In the previous situations, a nonpartitioned index will be created by default.

User response

If you do not want to create partitioned indexes on partitioned tables, specify the NOT PARTITIONED clause on the CREATE INDEX statement.

Related concepts

″Indexes on partitioned tables″ in Database Administration Concepts and Configuration Reference

Related reference

″CREATE INDEX ″ in SQL Reference, Volume 2

Some database manager configuration parameters have been changed

Version 9.7 contains a number of new and changed database manager configuration parameters.

The following database manager configuration parameters apply also to DB2 Connect.

New database manager configuration parameters

Due to new features and functionality, Version 9.7 contains a number of new configuration parameters.

Table 11. Summary of new Version 9.7 database manager configuration parameters

Parameter name	Description	Details
alternate_auth_enc	Alternate encryption algorithm for incoming connections at the server	Specifies the alternate encryption algorithm to be used to encrypt the user ID and password submitted to DB2 server for authentication when the authentication method negotiated between the DB2 client and the DB2 server is SERVER_ENCRYPT.
diagsize	Rotating diagnostic and administration notification logs	Controls the maximum size of diagnostic logs and administration notify logs.
ssl_cipherspecs	Supported cipher specifications at the server	Specifies the cipher suites that the server allows for incoming connection requests when using SSL protocol.
ssl_clnt_keydb	SSL key file path for outbound SSL connections at the client	Specifies the fully qualified file path of the key file to be used for SSL connection at the client-side.
ssl_clnt_stash	SSL stash file path for outbound SSL connections at the client	Specifies the fully qualified file path of the stash file to be used for SSL connections at the client-side.
ssl_svr_keydb	SSL key file path for incoming SSL connections at the server	Specifies a fully qualified file path of the key file to be used for SSL setup at server-side.
ssl_svr_label	Label in the key file for incoming SSL connections at the server	Specifies a label of the personal certificate of the server in the key database.
ssl_svr_stash	SSL stash file path for incoming SSL connections at the server	Specifies a fully qualified file path of the stash file to be used for SSL setup at server-side.
ssl_svcename	SSL service name	Specifies the name of the port that a database server uses to await communications from remote client nodes using SSL protocol.
ssl_versions	Supported SSL versions at the server	Specifies SSL and TLS versions that the server supports for incoming connection requests.

Changed database manager configuration parameters

The following database manager configuration parameters have changed behaviors in Version 9.7.

Table 12. Summary of database manager configuration parameters with changed behaviors

Parameter name	Description	Version 9.7 change
authentication and **srvcon_auth**	Authentication type configuration parameter and Authentication type for incoming connections at the server configuration parameter	If you enabled 256-bit AES encryption for user IDs and passwords, check the **alternate_auth_enc** parameter which allows you to specify an alternate encryption algorithm for user names and passwords. For more information, see the **alternate_auth_enc** parameter.

Related concepts

"Some database configuration parameters have been changed" on page 142

Related reference

″RESET DATABASE CONFIGURATION ″ in Command Reference

″Configuration parameters summary″ in Database Administration Concepts and Configuration Reference

NO FILE SYSTEM CACHING for table space containers is the default for General Parallel File System™ (GPFS)

Starting in Version 9.7, when the underlying file system is GPFS™, NO FILE SYSTEM CACHING is the default behavior for table space definition on a subset of platforms if you do not specify the FILE SYSTEM CACHING option on the CREATE TABLESPACE statement and on some of the table space definition parameters of the CREATE DATABASE command.

Details

For the CREATE DATABASE command, this behavior applies to the CATALOG, USER, non-SMS TEMPORARY table space definition parameters.

In previous releases, the default behavior was FILE SYSTEM CACHING on GPFS for all supported platforms. In Version 9.7, the default has been changed to NO FILE SYSTEM CACHING on a subset of AIX and Linux platforms. The new default specifies that I/O operations bypass the file system caching automatically.

User response

For details on the platforms that support NO FILE SYSTEM CACHING, see the "File system caching configurations" topic. To revert to buffered I/O behavior, specify the FILE SYSTEM CACHING attribute in the CREATE TABLESPACE statement, the ALTER TABLESPACE statement, or the CREATE DATABASE command.

Related concepts

″File system caching configurations″ in Database Administration Concepts and Configuration Reference

Related reference

″ALTER TABLESPACE ″ in SQL Reference, Volume 2

″CREATE TABLESPACE ″ in SQL Reference, Volume 2

Some registry and environment variables have changed

In Version 9.7, there are a number of changes to registry and environment variables.

New defaults

Table 13. Registry variables with new default values

Registry variable	Version 9.5 default setting	Version 9.7 default setting
DB2_LOGGER_NON_ BUFFERED_IO	OFF	Starting with Version 9.7, the default value for this variable is AUTOMATIC, which means that log files in the active log path might be accessed using non-buffered I/O. The database manager determines which log files benefit from using non-buffered I/O. In Version 9.5 Fix Pack 1 or later, the default was OFF and log files were accessed using only buffered I/O.

Table 14. Registry variables with new values

Registry variable	New values
DB2_EVMON_STMT_FILTER	This variable has new options that allow users to determine which rules to apply to which event monitors. Each option represents an integer value mapping to a specific SQL operation. These new options are also available starting in DB2 Version 9.5 Fix Pack 1.
DB2_SQLROUTINE_PREPOPTS	This variable has two new options: • **APREUSE**, which indicates whether the query compiler will attempt to reuse access plans. • **CONCURRENTACCESSRESOLUTION**, which specifies the concurrent access resolution to use for statements in the package

Table 14. Registry variables with new values (continued)

Registry variable	New values
DB2_WORKLOAD	This variable has two new values: CM and WC. These settings allow you to configure a set of registry variables in your database for applications provided by IBM Content Manager and IBM Websphere Commerce. The CM and WC values are also available starting in DB2 Version 9.5 Fix Pack 3 and Fix Pack 4, respectively.

Changed behaviors

Table 15. Registry variables with changed behaviors

Registry variable	Changed behavior
DB2_EVALUNCOMMITTED and **DB2_SKIPDELETED**	For statements operating under cursor stability isolation level with currently committed behavior enabled using the **cur_commit** database configuration parameter, these registry variables are in effect only when currently committed cannot be applied to a scan. Otherwise, the evaluation of predicates is performed on data retrieved by currently committed scans. If currently committed behavior was enabled using the BIND command or the PREPARE statement, these registry variables have no effect. For more information, see the **cur_commit** configuration parameter.
DB2_SERVER_ENCALG	The **DB2_SERVER_ENCALG** registry variable is deprecated. If the **alternate_auth_enc** database manager configuration parameter is set, its value has precedence over the **DB2_SERVER_ENCALG** value.
DB2_SKIPINSERTED	For statements operating under cursor stability isolation level with currently committed behavior enabled, this registry variable has no effect. For more information, see the **cur_commit** configuration parameter.

New variables

These environment and registry variables are new in Version 9.7:

Table 16. Environment and Registry variables added

Registry variable	Description
DB2_ATS_ENABLE	This registry variable enables or disables the administrative task scheduler.

Table 16. Environment and Registry variables added (continued)

Registry variable	Description
DB2_DDL_SOFT_INVAL	This registry variable enables *soft invalidation* of applicable database objects when they are dropped or altered, meaning that active access to an object that is being invalidated can continue.
DB2_FCM_SETTINGS	On Linux operating systems, you can set this registry variable with the FCM_MAXIMIZE_SET_SIZE token to preallocate a default 2 GB of space for the fast communication manager (FCM) buffer. The token must have a value of either YES or TRUE to enable this feature.
DB2_FORCE_OFFLINE_ADD_PARTITION	This environment variable allows you to specify whether add partition operations are to be performed offline or online. The default setting of FALSE indicates that DB2 partitions can be added without taking the database offline.
DB2_DEFERRED_PREPARE_SEMANTICS	This registry variable allows dynamic statements containing untyped parameter markers to use deferred prepare semantics. By default, this variable is set to YES so that any untyped parameter markers derive their data types and length attributes based on the input descriptor from subsequent OPEN or EXECUTE statements. In previous releases, compilation of such dynamic statements would have failed.
DB2_PMAP_COMPATIBILITY	This variable allows users to continue using the sqlugtpi and sqlugrpn APIs to return, respectively, the distribution information for a table, and the distribution map offset and database partition for a row. When this variable is set to OFF, the distribution map size for new or upgraded databases is increased to 32 768 entries and you have to use the new db2GetDistMap and db2GetRowPartNum APIs.

Table 16. Environment and Registry variables added (continued)

Registry variable	Description
DB2RESILIENCE	This environment variable controls whether DB2 data page read errors are tolerated, and activates extended trap recovery. It is set to ON by default. To revert to the behavior of previous releases and force the database manager to shutdown the instance, set the registry variable to OFF.

Related concepts

"Some registry and environment variables have been deprecated" on page 183

"Some registry and environment variables have been discontinued" on page 192

Primary and secondary log files use non-buffered I/O by default

In Version 9.7, primary and secondary recovery log files use non-buffered I/O automatically, eliminating the overhead incurred by the operating system when caching these log files.

Details

With the new behavior, the file system where the primary and secondary recovery logs reside should not be mounted with options to disable the file system cache buffering.

In previous releases, the default behavior for these log files was to use buffered I/O. You can revert to the behavior of the previous releases by setting the **DB2_LOGGER_NON_BUFFERED_IO** registry variable to OFF.

In certain situations, the new behavior might degrade the performance of log disk I/O response times resulting in longer commit times. There might also be performance impact for long rollback operations.

User response

You can address performance degradation for longer commit times by ensuring that the number of physical disk spindles for the logger file system matches the desired performance levels. In addition, you can improve performance by enabling the storage controller write caching mechanisms, provided that these mechanisms meet system durability requirements so that your system can recover committed transaction updates if either the system or the storage media fails.

You can address rollback performance issues by tuning the **logbufsz** database configuration parameter to ensure that log data required for rollforward operations is in the log buffer instead of having to perform physical read I/O on the disk.

The CONCURRENTDBCOORDACTIVITIES threshold has been changed

To reduce the chance of creating inadvertent deadlock scenarios, the behavior of the CONCURRENTDBCOORDACTIVITIES threshold has been changed.

Details

When creating a CONCURRENTDBCOORDACTIVITIES threshold, inadvertent queue deadlock scenarios can result. A deadlock scenario occurs when a concurrency limit has been reached and all applications that have issued the activities that are holding the tickets then try to issue one or more additional activities. Those additional activities get queued because there are no more tickets available, thereby stopping the applications from proceeding. For example, if the concurrency threshold permits only one activity to be processed at a time and a single application opens one cursor and then tries to issue another activity of any type. The cursor opened by the application acquires the single ticket. The second activity gets queued since there are no more available tickets, thus putting the application into a deadlock.

The chance of creating inadvertent deadlock scenarios has been reduced by changes to the behavior of the CONCURRENTDBCOORDACTIVITIES threshold, which now controls fewer types of activities than before:

- CALL statements are no longer controlled by the threshold, but all nested child activities started within the called routine continue to be under threshold control. Note that both anonymous blocks and autonomous routines are classified as CALL statements.
- User defined functions (UDFs) continue to be under threshold control, but child activities nested within UDFs are no longer controlled. If an autonomous routine is called from within a user defined function, neither the autonomous routine and nor any child activities of the autonomous routine are under threshold control.
- Trigger actions that invoke CALL statements and the child activities of these CALL statements are no longer under threshold control. Note that the insert, update or delete statements themselves that can cause a trigger activate continue to be under threshold control.

The behavior of the CONCURRENTDBCOORDACTIVITIES threshold for all other types of activities remains unchanged.

User response

Continue to use the CONCURRENTDBCOORDACTIVITIES threshold to impose control over the maximum number of recognized coordinator activities that can run concurrently across all database partitions. If you create a CONCURRENTDBCOORDACTIVITIES threshold with a very low concurrency limit, it is still possible to create a deadlock scenario unless an additional step is taken. To avoid deadlock scenarios, you should always use the CONCURRENTDBCOORDACTIVITIES threshold in conjunction with an ACTIVITYTOTALTIME threshold to prevent activities from remaining queued indefinitely.

DESCRIBE command lists information about additional index types

By default, the DESCRIBE command with the **INDEXES FOR TABLE** parameter now lists information about the system-generated XML regions index and XML path indexes, and DB2 Text Search indexes, in addition to information about relational indexes and indexes over XML data.

Details

If you specify the **INDEXES FOR TABLE** parameter with the SHOW DETAIL clause, more information is listed for all the types of indexes.

User response

Because the index information displayed by the DESCRIBE command with the **INDEXES FOR TABLE** parameter contains new columns, you must change tools that depend on the output to parse the new text.

Related reference

″DESCRIBE ″ in Command Reference

Database setup and product installation changes summary

Version 9.7 includes changed functionality that affects how you install and setup DB2 databases.

You can upgrade DB2 server or client copies to DB2 Version 9.7 from DB2 Version 9.5, DB2 Version 9.1, or DB2 UDB Version 8. DB2 Version 9.7 is a new release and you cannot apply a fix pack to upgrade from Version 9.5 or Version 9.1 to Version 9.7. If you have a Version 7 or earlier copy installed, migrate it first to DB2 UDB Version 8.

To learn details, limitations of the upgrade process, and possible issues that you need to be aware of, review “Upgrade essentials for DB2 servers” and “Upgrade essentials for clients” in *Upgrading to DB2 Version 9.7*.

Upgrading your DB2 servers and clients to Version 9.7 might require that you also upgrade your database applications and routines. Review “Upgrade essentials for database applications” and “Upgrade essentials for routines” in *Upgrading to DB2 Version 9.7* to help you determine whether there is any upgrade impact.

Related reference

″Installation requirements for DB2 database products″ in Installing DB2 Servers

License enforcement policies list has been updated

To reflect the Version 9.7 product packaging, the list of license enforcement policies includes row level compression and index compression, and no longer includes the pureXML feature.

Details

License enforcement policies are configured for your DB2 database products using the db2licm command with the **-e** option.

If you choose to use a hard-stop license enforcement policy for your DB2 database product, the database manager will check for licensing compliance when users attempt to use row level compression and index compression. If the appropriate licenses have not been applied, a SQL8029N message will be returned and the attempted action will not be allowed.

User response

Apply the appropriate licenses for row level compression and index compression.

Some database configuration parameters have been changed

Version 9.7 contains a number of new and changed database configuration parameters.

New database configuration parameters

Due to new features and functionality, Version 9.7 contains a number of new database configuration parameters.

Table 17. New Version 9.7 database configuration parameters

Parameter name	Description	Details
auto_reval	Automatic revalidation and invalidation	This configuration parameter controls the revalidation and invalidation semantics. This parameter is dynamic, meaning that a change in its value takes effect immediately. You do not have to reconnect to the database for the change to take effect.
blocknonlogged	Block non-logged activity	This configuration parameter prevents the creation of tables that allow non-logged activity.
cur_commit	Currently committed	This configuration parameter controls the behavior of cursor stability (CS) scans.
date_compat	Date compatibility	This parameter indicates whether the DATE compatibility semantics associated with the TIMESTAMP(0) data type are applied to the connected database.
dec_to_char_fmt	Decimal to character function configuration parameter	This configuration parameter controls the result of the CHAR scalar function and the CAST specification for converting decimal to character values.
mon_act_metrics	Monitoring activity metrics	These parameters control the collection of metrics and event monitor data at the database level including the new lock event monitor. During database upgrade, these parameters are set to NONE, except for **mon_deadlock** which is set to WITHOUT_HIST and **mon_lw_thresh** which is set to 5 000 000, so that there is no change in behavior from previous releases.
mon_deadlock	Monitoring deadlock	
mon_locktimeout	Monitoring lock timeout	
mon_lockwait	Monitoring lock wait	
mon_lw_thresh	Monitoring lock wait threshold	
mon_obj_metrics	Monitoring object metrics	
mon_req_metrics	Monitoring request metric	
mon_uow_data	Monitoring unit of work events	
stmt_conc	Statement concentrator	This configuration parameter enables statement concentration for dynamic statements. The setting in the database configuration is used only when the client does not explicitly enable or disable statement concentrator.

Changed database configuration parameters

The following table lists the database configuration parameters with changes to their default values.

Table 18. Database configuration parameters with changed default values

Parameter name	Description	Version 9.5 default value	Version 9.7 default value
logbufsz	Log buffer size	8 pages (each 4KB)	256 pages (each 4KB)

The following database configuration parameters have changed behaviors or have new ranges in Version 9.7.

Table 19. Database configuration parameters with changed behaviors or new ranges

Parameter name	Description	Version 9.7 change
applheapsz	Application heap size	Due to optimization enhancements to match MQTs, the requirement for application heap has increased. If this parameter is set to AUTOMATIC, this setting accounts for the new requirements. If you cannot set this parameter to AUTOMATIC or increase its value, reduce the number of MQTs considered for a given query by using optimization profiles. For more information, see “Anatomy of an optimization profile” in *Troubleshooting and Tuning Database Performance*.
dbheap	Database heap	The database manager can now determine when to apply row compression to temporary tables that meet certain criteria to improve query performance. Memory allocated for database heap is used to create the compression dictionary and released once the dictionary is created. If you are using row compression and temporary tables eligible for compression, ensure that you have enough space to create the dictionary by setting the **dbheap** parameter to AUTOMATIC. For details about temporary table compression, see “Table compression” in *Database Administration Concepts and Configuration Reference*.
locklist	Maximum storage for lock list	The limit for this parameter is now 134,217,728 pages (4 KB).
logbufsz	Log buffer size	A log sequence number (LSN) uses now 8 bytes. In previous releases, LSN was 6 bytes in length. You might need to increase the value of this parameter according to your database logging activity. For more information, see “Maximum limit of log sequence numbers has increased” on page 153. The maximum limit for **logbufsz** has been changed to 131 070. The maximum limit for **logfilsiz** has been changed to 1 048 572.
logfilsiz	Size of log files	
logprimary	Number of primary log files	
pckcachesz	Package cache size	To support XML Explain, package cache memory requirements have increased from 10 to 25 percent. The impact from the database upgrade should be minimal because of the small size of this cache. By setting this parameter to AUTOMATIC, the new requirements are taken into account. For upgraded databases, the INLINE LENGTH default value is the maximum size of the LOB descriptor. LOB data is inlined when the length of the LOB data plus the overhead do not exceed the INLINE LENGTH value. Therefore, if the LOB data length plus the overhead is less than the LOB descriptor size for the LOB column, the LOB data is implicitly inlined in a table row after the database upgrade. Storing LOB data as inlined might require that you increase the **pckcachesz** database configuration parameter. By setting this parameter to AUTOMATIC, the new requirements are taken into account. The maximum limit for **pckcachesz** on 64-bit operating systems has been changed to 2 147 483 646.

Deprecated configuration parameters

Table 20. Summary of deprecated database configuration parameters

Parameter name	Description	Details and resolution
dyn_query_mgmt	Dynamic SQL and XQuery query management	This configuration parameter is deprecated because it is Query Patroller specific. With the new workload management features introduced in DB2 Version 9.5, Query Patroller and its components have been deprecated in Version 9.7 and might be removed in a future release.

Related concepts

“Some database manager configuration parameters have been changed” on page 133

Related reference

″RESET DATABASE CONFIGURATION ″ in Command Reference

″Configuration parameters summary″ in Database Administration Concepts and Configuration Reference

INTERACTIVE response file keyword has been changed

The existing response file keyword, INTERACTIVE, no longer prompts for the location of the installation package.

Details

The INTERACTIVE keyword applies to Linux and UNIX operating systems only. In previous releases, if the INTERACTIVE keyword was set to YES, the user was prompted for either the location of the installation package, or, the location of the national language package. In Version 9.7, the INTERACTIVE keyword prompts for the location of the national language package only. The installation images are now available on one DVD, therefore, this keyword no longer requires a prompt for the location of the installation package. Prompting occurs if the INTERACTIVE keyword was set to YES, and a national language DVD is required.

User response

You do not have to change your applications or scripts.

Related concepts

“UPGRADE_PRIOR_VERSIONS response file keyword has been added” on page 121

“Some response file keywords have been deprecated” on page 185

Related reference

″Response file keywords″ in Installing DB2 Servers

Registry files have been removed from the DB2 installation path

The location of instance information and global registry information has been changed. Starting with DB2 Version 9.7, the profiles.reg and default.env files are removed from the DB2 installation path.

Details

In DB2 Version 9.5, the DB2 Instance Profile Registry was located in the profiles.reg file, and the DB2 Global-Level Profile Registry was located in the default.env file.

These files were in the DB2 installation path.

User response

DB2 instance information and global registry information is stored in the global registry (global.reg).

Related tasks

″Setting environment variables on Linux and UNIX operating systems″ in Database Administration Concepts and Configuration Reference

DB2 Text Search installation has changed

In Version 9.7, to install the DB2 Text Search product, you must select **Custom** as the type of DB2 installation. In addition, you no longer have to specify two commas before the Text Search instance services port number when using certain DB2 commands on the Windows operating systems.

Details

DB2 Text Search is no longer installed automatically if you select **Typical** when installing the DB2 product.

On Windows operating systems, the syntax for the Text Search instance services port number has been simplified for the following commands:

- db2icrt (Create instance)
- db2imigr (Migrate instance)
- db2iupdt (Update instances)

In addition, the new Version 9.7 db2iupgrade command uses the simplified syntax. The simplified syntax is as follows:

```
/j "TEXT_SEARCH,portnumber"
```

User response

To install DB2 Text Search in Version 9.7, select **Custom** when installing the DB2 product. Also, add the COMP=TEXT_SEARCH and the CONFIGURE_TEXT_SEARCH=YES keywords to existing scripts and response files.

If you have existing scripts on Windows operating systems that use the db2icrt, the db2imigr, or the db2iupdt command, remove the extra comma at the beginning of the Text Search instance services port number.

Security changes summary

Version 9.7 includes changed functionality that affects the scope and abilities of SYSADM, SECADM and DBADM authority levels, SSL configuration, and other features.

System administrator (SYSADM) authority scope has changed

In DB2 Version 9.7, the authorization model has been updated to clearly separate the duties of the system administrator, the database administrator, and the security administrator. As part of this enhancement, the abilities given by the SYSADM authority have been reduced.

Details

The changes for the SYSADM authority are as follows:

- A user who holds SYSADM authority no longer has implicit DBADM authority and therefore has limited capabilities compared to those available in Version 9.5. However, the UPGRADE DATABASE command and the RESTORE DATABASE command (for a downlevel database) grants DBADM authority to the SYSADM group. Privileges associated with groups are not considered for authorization when a user creates views, triggers, materialized query tables (MQTs), packages and SQL routines. Given these restrictions associated with groups, even though the upgrade process grants DBADM authority to the SYSADM group, the upgrade process alone does not ensure that every user with SYSADM authority in Version 9.5 will have the exact same capabilities in Version 9.7. For a member of the SYSADM group to be certain to retain the same privileges as in Version 9.5, they must be directly granted DBADM authority with DATAACCESS and ACCESSCTRL authorities, or must possess these authorities through membership of a role.
- If a user holding SYSADM authority creates a database, the user is automatically granted DATAACCESS, ACCESSCTRL, SECADM and DBADM authority for that database, which gives the user the same abilities as in Version 9.5.
- A user who holds SYSADM authority is no longer able to grant any authorities or privileges, except table space privileges.

User response

For a user holding SYSADM authority to obtain the same capabilities as in Version 9.5 (other than the ability to grant SECADM authority), the security administrator must explicitly grant the user DBADM authority and grant the user the new DATAACCESS and ACCESSCTRL authorities. The new authorities can be granted by using the GRANT DBADM ON DATABASE statement with the WITH DATAACCESS and WITH ACCESSCTRL options of that statement, which are default options. The DATAACCESS authority is the authority that allows access to data within a specific database, and the ACCESSCTRL authority is the authority that allows a user to grant and revoke privileges within a specific database.

For the user holding SYSADM authority to also be able to grant SECADM authority, the security administrator must grant the user SECADM authority as well. However, holding SECADM authority allows the user to perform more actions than the user could as a Version 9.5 system administrator. For example, the user can create objects such as roles, trusted contexts, and audit policies.

Tip: In addition to considering how these SYSADM authority changes impact your security implementation, you should also review the new capabilities of the database administrator (who holds DBADM authority) and the security administrator (who holds SECADM authority) , and the new authorities introduced in DB2 Version 9.7, so that you can decide how to organize responsibilities within your system. DB2 Version 9.7 introduces the following new authorities in addition to DATAACCESS and ACCESSCTRL:

- WLMADM, for managing workloads
- SQLADM, for tuning SQL statements
- EXPLAIN, for using the explain facility with SQL statements

These new authorities allow you to grant users responsibilities without granting them DBADM authority or privileges on base tables, which would give those users more privileges than they need to do their work.

Considerations for the Windows LocalSystem account

On Windows systems, when the **sysadm_group** database manager configuration parameter is not specified, the LocalSystem account is considered a system administrator (holding SYSADM authority). Any DB2 application that is run by LocalSystem is affected by the change in scope of SYSADM authority in Version 9.7. These applications are typically written in the form of Windows services and run under the LocalSystem account as the service logon account. If there is a need for these applications to perform database actions that are no longer within the scope of SYSADM, you must grant the LocalSystem account the required database privileges or authorities. For example, if an application requires database administrator capabilities, grant the LocalSystem account DBADM authority using the GRANT (Database Authorities) statement. Note that the authorization ID for the LocalSystem account is SYSTEM.

Related concepts

″Authorization, privileges, and object ownership″ in SQL Reference, Volume 1

″Windows LocalSystem account support″ in Database Security Guide

“DB2 authorization model has been enhanced to allow separation of duties” on page 63

″Authorities overview″ in Database Security Guide

Security administrator (SECADM) abilities have been extended

In DB2 Version 9.7, the authorization model has been updated to clearly separate the duties of the system administrator, the database administrator, and the security administrator. As part of this enhancement, the abilities given by the SECADM authority have been extended.

Details

The changes for the SECADM authority are as follows:

- A user who holds SECADM authority can now grant and revoke all authorities and privileges, including DBADM and SECADM.
- The security administrator can now grant SECADM authority to roles and groups. In Version 9.5, SECADM could be granted only to a user.
- The security administrator can delegate responsibility to run the audit stored procedures and table functions (AUDIT_ARCHIVE, AUDIT_LIST_LOGS, and AUDIT_DELIM_EXTRACT) by granting another user EXECUTE privilege on them.

User response

The security administrator can allow another user to grant and revoke authorities and privileges by granting that other user the new ACCESSCTRL authority. However, only the security administrator can grant SECADM, DBADM, and ACCESSCTRL authority. Also, only the security administrator can grant the new authority DATAACCESS, which enables a user to access data within a specific database.

In addition to considering how these SECADM authority changes impact your security implementation, you should also review the new capabilities of the system administrator (who holds SYSADM authority) and the database administrator (who holds DBADM authority) , and the new authorities introduced in DB2

Version 9.7, so that you can decide how you to organize responsibilities within your system. DB2 Version 9.7 introduces the following new authorities in addition to DATAACCESS and ACCESSCTRL:

- WLMADM, for managing workloads
- SQLADM, for tuning SQL statements
- EXPLAIN, for using the explain facility with SQL statements

These new authorities allow you to grant users responsibilities without granting them DBADM authority or privileges on base tables, which would give those users more privileges than they need to do their work.

Related concepts

″Authorization, privileges, and object ownership″ in SQL Reference, Volume 1

"DB2 authorization model has been enhanced to allow separation of duties" on page 63

″Authorities overview″ in Database Security Guide

Database administrator (DBADM) authority scope has changed

In DB2 Version 9.7, the authorization model has been updated to clearly separate the duties of the system administrator, the database administrator, and the security administrator. As part of this enhancement, the abilities given to the DBADM authority have changed.

Details

The changes for the DBADM authority are as follows:

- DBADM authority no longer necessarily includes the ability to access data and to grant and revoke privileges for a database.
- Granting DBADM authority no longer additionally grants the following separate database authorities because they are already implicitly vested in the DBADM authority level.
  - BINDADD
  - CONNECT
  - CREATETAB
  - CREATE_EXTERNAL_ROUTINE
  - CREATE_NOT_FENCED_ROUTINE
  - IMPLICIT_SCHEMA
  - QUIESCE_CONNECT
  - LOAD

User response

The new DATAACCESS authority provides the ability to access data in a database, and the new ACCESSCTRL authority provides the ability to grant and revoke privileges and authorities. These authorities are granted by default when a security administrator grants DBADM authority. Also, the security administrator can use the following options of the GRANT DBADM ON DATABASE statement to provide or not provide the ACCESSCTRL and DATAACCESS authorities:

- WITH ACCESSCTRL
- WITHOUT ACCESSCTRL

- WITH DATAACCESS
- WITHOUT DATAACCESS

Tip: In addition to considering how these DBADM authority changes impact your security implementation, you should also review the new capabilities of the system administrator (who holds SYSADM authority) and security administrator (who holds SECADM authority) , and the new authorities introduced in DB2 Version 9.7, so that you can decide how you to organize responsibilities within your system. DB2 Version 9.7 introduces the following new authorities in addition to DATAACCESS and ACCESSCTRL:

- WLMADM, for managing workloads
- SQLADM, for tuning SQL statements
- EXPLAIN, for using the explain facility with SQL statements

These new authorities allow you to grant users responsibilities without granting them DBADM authority or privileges on base tables, which would give those users more privileges than they need to do their work.

Related concepts

″Authorization, privileges, and object ownership″ in SQL Reference, Volume 1

"DB2 authorization model has been enhanced to allow separation of duties" on page 63

″Authorities overview″ in Database Security Guide

SSLconfig.ini and SSLClientconfig.ini files replaced with new database manager configuration parameters

You no longer need to use the SSLconfig.ini and SSLClientconfig.ini configuration files to set up SSL support. The parameters that you used to set in these files have been replaced with database manager configuration parameters.

Details

The new database manager configuration parameters for server-side SSL support are as follows:

- **ssl_svr_keydb** specifies the fully qualified path of the key database file.
- **ssl_svr_stash** specifies the fully qualified path of the stash file that holds the encrypted password to the key database.
- **ssl_svr_label** specifies the label of the digital certificate of the server in the key database.
- **ssl_svcename** specifies the port that the database server uses to await communications from remote clients using the SSL protocol.
- **ssl_cipherspecs** (optional) specifies the cipher suites that the server supports.
- **ssl_versions** (optional) specifies the SSL and TLS versions that the server supports.

The new database manager configuration parameters for client-side SSL support are as follows:

- **ssl_clnt_keydb** specifies the fully qualified path of the key database file on the client.
- **ssl_clnt_stash** specifies the fully qualified path of the stash file on the client.

User response

To set up SSL support, set values for the new database manager configuration parameters.

The following tables show how the parameters in the SSLconfig.ini and SSLClientconfig.ini files map to these new database manager configuration parameters. The **ssl_cipherspecs** and **ssl_versions** parameters do not have equivalent parameters in these files; they provide new configuration options.

Table 21. Mapping of server-side SSL support parameters to new database manager configuration parameters

Version 9.5 SSLconfig.ini parameters	Version 9.7 database manager configuration parameters
DB2_SSL_KEYSTORE_FILE	ssl_svr_keydb
DB2_SSL_KEYSTORE_PW	ssl_svr_stash
DB2_SSL_KEYSTORE_LABEL	ssl_svr_label
DB2_SSL_LISTENER	ssl_svcename

The **ssl_svr_stash** database manager configuration parameter is not exactly equivalent to the **DB2_SSL_KEYSTORE_PW** parameter. The **ssl_svr_stash** configuration parameter points to a stash file that holds the encrypted password to a key database, whereas the **DB2_SSL_KEYSTORE_PW** parameter specifies the password itself.

Table 22. Mapping of client-side SSL support parameters to new database manager configuration parameters

Version 9.5 SSLClientconfig.ini parameters	Version 9.7 database manager configuration parameters
DB2_SSL_KEYSTORE_FILE	ssl_clnt_keydb
DB2_SSL_KEYRING_STASH_FILE	ssl_clnt_stash

Related concepts

"SSL client support expanded and configuration simplified" on page 66

Audit stored procedures and table functions now require only the EXECUTE privilege

In Version 9.7, the security administrator (who holds SECADM authority) can grant the EXECUTE privilege on the audit stored procedures and table functions. Only the security administrator has the ability to grant EXECUTE on these routines.

Details

Prior to Version 9.7, only the security administrator could run the following stored procedures and table functions:

- The AUDIT_ARCHIVE stored procedure and table function
- The AUDIT_LIST_LOGS table function
- The AUDIT_DELIM_EXTRACT stored procedure

Resolution

In Version 9.7, you can run the audit stored procedures and table functions if you have been granted the EXECUTE privilege on them.

Net Search Extender command authorizations have changed

Version 9.7 includes authorization changes that affect the scope and abilities of SYSADM, SECADM and DBADM authority levels and therefore impact Net Search Extender command execution.

Details

Starting in Version 9.7, the instance owner must hold both the DBADM and DATAACESS authorities, otherwise the Net Search Extender commands will fail even if the user has the correct authorities and privileges.

In addition, the authorities and privileges required for running the following Net Search Extender commands have changed as follows:

Table 23. Authorization changes for Net Search Extender commands

Command	Version 9.5 authorization	Version 9.7 authorization
ACTIVATE CACHE	CONTROL privilege on the index table	CONTROL privilege on the index table or DBADM authority
ALTER	CONTROL privilege on the index table	CONTROL privilege on the index table or DBADM authority
CLEAR EVENTS	CONTROL privilege on the index table	CONTROL privilege on the index table or DBADM authority
CONTROL	Instance owner must hold SYSADM authority	Instance owner must hold DBADM with DATAACCESS authority
CREATE INDEX	CONTROL privilege on the index table	One of the following authority levels is required: • CONTROL privilege on the index table • INDEX privilege on the index table and either IMPLICIT_SCHEMA authority on the database or CREATEIN privilege on the index table schema • DBADM authority
DB2EXTHL	CONNECT to DB privilege	Instance owner must hold DBADM with DATAACCESS authority
DEACTIVATE CACHE	CONTROL privilege on the index table	CONTROL privilege on the index table or DBADM authority
DISABLE	DBADM authority	DBADM authority
DROP INDEX	CONTROL privilege on the index table or DBADM authority	CONTROL privilege on the index table or DBADM authority
ENABLE	DBADM with SYSADM authority	DBADM authority
UPDATE	CONTROL privilege on the index table	CONTROL privilege on the index table or DATAACCESS authority

User response

Ensure that the instance owner has both DBADM and DATAACCESS authorities and ensure that you have the appropriate authority levels and privileges before running the Net Search Extender commands.

DB2 Text Search command and stored procedure authorizations have changed

Version 9.7 includes authorization changes that affect the scope and abilities of SYSADM, SECADM and DBADM authority levels and therefore impact Text Search commands and stored procedure execution.

Details

Starting in Version 9.7, the instance owner must hold both the DBADM and DATAACESS authorities, otherwise the DB2 Text Search commands and stored procedures will fail even if the user has the correct authorities and privileges.

In addition, the authorities and privileges required for running the DB2 Text Search commands and stored procedures have changed as follows:

Table 24. Authorization changes for the db2ts command

db2ts command	Version 9.5 authorization	Version 9.7 authorization
ALTER	CONTROL privilege on the index table	CONTROL privilege on the index table or DBADM authority
CLEANUP	Instance owner	Instance owner must hold DBADM with DATAACCESS authority
CLEAR COMMAND LOCKS	CONTROL privilege on the index table, or DBADM or SYSADM if no index is specified.	CONTROL privilege on the index table, or DBADM authority if no index is specified
CLEAR EVENTS	CONTROL privilege on the index table	CONTROL privilege on the index table or DBADM authority
CREATE INDEX	CONTROL privilege on the index table	One of the following authority levels is required: • CONTROL privilege on the index table • INDEX privilege on the index table and either IMPLICIT_SCHEMA authority on the database or CREATEIN privilege on the index table schema • DBADM authority
DISABLE	DBADM or SYSADM authority	DBADM authority
DROP INDEX	CONTROL privilege on the index table	CONTROL privilege on the index table or DBADM authority
ENABLE	SYSADM authority	DBADM authority
UPDATE	CONTROL privilege on the index table	CONTROL privilege on the index table or DATAACCESS authority

Important: You must grant the EXECUTE privilege to PUBLIC for all the DB2 Text Search stored procedures.

Table 25. Authorization changes for the DB2 Text Search stored procedures

Stored procedure	Version 9.5 authorization	Version 9.7 authorization
SYSTS_ALTER	CONTROL privilege on the index table	CONTROL privilege on the index table or DBADM authority
SYSTS_ADMIN_CMD	authorization requirements are the same as those listed for the invoked command	authorization requirements are the same as those listed for the invoked command
SYSTS_CLEAR_COMMAND_LOCKS	If an index is specified, CONTROL privilege on the index, else if no index is specified, DBADM or SYSADM authority	If an index is specified, CONTROL privilege on the index table else if no index is specified, DBADM with authority
SYSTS_CREATE	CONTROL privilege on the index table	One of the following authority levels is required: • CONTROL privilege on the index table • INDEX privilege on the index table with either the IMPLICIT_SCHEMA authority on the database or the CREATEIN privilege on the index table schema • DBADM authority
SYSTS_CLEAR_EVENTS	CONTROL privilege on the index table	CONTROL privilege on the index table or DBADM authority
SYSTS_DISABLE	DBADM or SYSADM authority	DBADM authority
SYSTS_DROP	CONTROL privilege on the index table	CONTROL privilege on the index table or DBADM authority
SYSTS_ENABLE	SYSADM authority	DBADM authority
SYSTS_UPDATE	CONTROL privilege on the index table	CONTROL privilege on the index table or DATAACCESS authority

User response

Ensure that the instance owner has both DBADM and DATAACCESS authorities.

Ensure that you have the appropriate authority levels and privileges before running the DB2 Text Search commands or stored procedures, and that you have granted the EXECUTE privilege to PUBLIC for all the DB2 Text Search stored procedures.

Application development changes summary

Version 9.7 includes changed functionality that affects how you develop applications.

Maximum limit of log sequence numbers has increased

Individual log records within a database are identified by their log sequence number (LSN). In this release, the upper limit for LSNs has increased. The size of the LSN has been changed from six bytes to eight bytes.

Details

A new API data type, db2LSN, has been defined in db2ApiDf.h in support of the new LSN size.

For information on what happens between new and old client and server combinations, see “Log sequence number changes affecting API and application behavior”.

User response

There is no down level API support for the log read APIs affected by this change. You need to update existing applications using log read APIs (db2ReadLog and db2ReadLogNoConn APIs) to use the new release libraries after upgrading the database server. Clients must also be upgraded to the new release to use the new libraries.

You must also change applications to use the new LSN data structure differences present in the log flow that is returned in the log buffer during a log read API operation.

The error message, SQL2032N, is returned to indicate an unsupported down level API call.

Related concepts

“Data replication source tables can be compressed” on page 7

″Log sequence number changes affecting API and application behavior″ in Administrative API Reference

Some system catalog views, system-defined administrative routines and views have been added and changed

To support new features in Version 9.7, system catalog views, system built-in routines, administrative routines and views have been added and modified.

System catalog view changes

The following system catalog views have changed in Version 9.7. Most modifications to catalog views consist of new columns, changed descriptions, changed column data types, and increased column lengths.

- SYSCAT.ATTRIBUTES
- SYSCAT.BUFFERPOOLS
- SYSCAT.CASTFUNCTIONS
- SYSCAT.COLUMNS
- SYSCAT.CONSTDEP
- SYSCAT.DATAPARTITIONS
- SYSCAT.DATATYPES
- SYSCAT.DBAUTH
- SYSCAT.HISTOGRAMTEMPLATEUSE
- SYSCAT.INDEXDEP
- SYSCAT.INDEXES
- SYSCAT.INDEXEXTENSIONDEP
- SYSCAT.INVALIDOBJECTS
- SYSCAT.PACKAGEDEP
- SYSCAT.PACKAGES
- SYSCAT.ROUTINEDEP
- SYSCAT.ROUTINEPARMS

- SYSCAT.ROUTINES
- SYSCAT.SECURITYPOLICIES
- SYSCAT.SEQUENCES
- SYSCAT.SERVICECLASSES
- SYSCAT.TABDEP
- SYSCAT.TABDETACHEDDEP
- SYSCAT.TABLES
- SYSCAT.TABLESPACES
- SYSCAT.THRESHOLDS
- SYSCAT.TRIGDEP
- SYSCAT.VARIABLEDEP
- SYSCAT.VARIABLES
- SYSCAT.WORKCLASSES
- SYSCAT.WORKLOADS
- SYSCAT.XSROBJECTDEP
- SYSSCAT.COLGROUPS
- SYSSCAT.COLUMNS
- SYSSCAT.INDEXES

The following system catalog views have been added in Version 9.7:

- SYSCAT.CONDITIONS
- SYSCAT.DATATYPEDEP
- SYSCAT.INDEXPARTITIONS
- SYSCAT.INVALIDOBJECTS
- SYSCAT.MODULEAUTH
- SYSCAT.MODULEOBJECTS
- SYSCAT.MODULES
- SYSCAT.ROWFIELDS
- SYSCAT.XMLSTRINGS
- SYSCAT.XSROBJECTDETAILS

System-defined administrative views and routine changes

The following administrative views and routines have changed in Version 9.7.

- ADMIN_CMD procedure
- ADMINTABCOMPRESSINFO administrative view and ADMIN_GET_TAB_COMPRESS_INFO table function
- ADMINTABINFO administrative view and ADMIN_GET_TAB_INFO_V97 table function
- AUTH_LIST_AUTHORITIES_FOR_AUTHID table function
- DBMCFG administrative view
- REBIND_ROUTINE_PACKAGE routine
- REORGCHK_IX_STATS
- SNAPAPPL_INFO administrative view and SNAP_GET_APPL_INFO_V95 table function
- SNAPSTORAGE_PATHS administrative view and SNAP_GET_STORAGE_PATHS_V97 table function

- SNAPTBSP_PART administrative view and SNAP_GET_TBSP_PART_V97 table function
- WLM_GET_SERVICE_SUBCLASS_STATS_V97 table function

The following ADMIN_CMD stored procedure and associated administrative SQL routines have been added:

- ADMIN_EST_INLINE_LENGTH
- ADMIN_GET_INDEX_COMPRESS_INFO
- ADMIN_GET_INDEX_INFO
- ADMIN_GET_TAB_COMPRESS_INFO_V97
- ADMIN_GET_TEMP_COLUMNS
- ADMIN_GET_TEMP_TABLES
- ADMIN_IS_INLINED
- ADMIN_REVALIDATE_DB_OBJECTS

The following miscellaneous routines and views:

- ADMIN_MOVE_TABLE
- ADMIN_MOVE_TABLE_UTIL

The following SQL procedures routines have been added:

- ALTER_ROUTINE_PACKAGE

The following common SQL API procedures have been added:

- CANCEL_WORK
- GET_CONFIG
- GET_MESSAGE
- GET_SYSTEM_INFO
- SET_CONFIG

The following system-defined modules routines have been added:

- DBMS_ALERT.REGISTER
- DBMS_ALERT.REMOVE
- DBMS_ALERT.REMOVEALL
- DBMS_ALERT.SET_DEFAULTS
- DBMS_ALERT.SIGNAL
- DBMS_ALERT.WAITANY
- DBMS_ALERT.WAITONE
- DBMS_JOB.BROKEN
- DBMS_JOB.CHANGE
- DBMS_JOB.INTERVAL
- DBMS_JOB.NEXT_DATE
- DBMS_JOB.REMOVE
- DBMS_JOB.RUN
- DBMS_JOB.SUBMIT
- DBMS_JOB.WHAT
- DBMS_LOB.APPEND
- DBMS_LOB.CLOSE

- DBMS_LOB.COMPARE
- DBMS_LOB.CONVERTTOBLOB
- DBMS_LOB.CONVERTTOCLOB
- DBMS_LOB.COPY
- DBMS_LOB.ERASE
- DBMS_LOB.GET_STORAGE_LIMIT
- DBMS_LOB.GETLENGTH
- DBMS_LOB.INSTR
- DBMS_LOB.ISOPEN
- DBMS_LOB.OPEN
- DBMS_LOB.READ
- DBMS_LOB.SUBSTR
- DBMS_LOB.TRIM
- DBMS_LOB.WRITE
- DBMS_LOB.WRITEAPPEND
- DBMS_OUTPUT.DISABLE
- DBMS_OUTPUT.ENABLE
- DBMS_OUTPUT.GET_LINE
- DBMS_OUTPUT.GET_LINES
- DBMS_OUTPUT.NEW_LINE
- DBMS_OUTPUT.PUT
- DBMS_OUTPUT.PUT_LINE
- DBMS_PIPE.CREATE_PIPE
- DBMS_PIPE.NEXT_ITEM_TYPE
- DBMS_PIPE.PACK_MESSAGE
- DBMS_PIPE.PACK_MESSAGE_RAW
- DBMS_PIPE.PURGE
- DBMS_PIPE.RECEIVE_MESSAGE
- DBMS_PIPE.REMOVE_PIPE
- DBMS_PIPE.RESET_BUFFER
- DBMS_PIPE.SEND_MESSAGE
- DBMS_PIPE.UNIQUE_SESSION_NAME
- DBMS_PIPE.UNPACK_MESSAGE
- DBMS_SQL.BIND_VARIABLE_BLOB
- DBMS_SQL.BIND_VARIABLE_CHAR
- DBMS_SQL.BIND_VARIABLE_CLOB
- DBMS_SQL.BIND_VARIABLE_DATE
- DBMS_SQL.BIND_VARIABLE_DOUBLE
- DBMS_SQL.BIND_VARIABLE_INT
- DBMS_SQL.BIND_VARIABLE_NUMBER
- DBMS_SQL.BIND_VARIABLE_RAW
- DBMS_SQL.BIND_VARIABLE_TIMESTAMP
- DBMS_SQL.BIND_VARIABLE_VARCHAR
- DBMS_SQL.CLOSE_CURSOR
- DBMS_SQL.COLUMN_VALUE_BLOB

- DBMS_SQL.COLUMN_VALUE_CHAR
- DBMS_SQL.COLUMN_VALUE_CLOB
- DBMS_SQL.COLUMN_VALUE_DATE
- DBMS_SQL.COLUMN_VALUE_DOUBLE
- DBMS_SQL.COLUMN_VALUE_INT
- DBMS_SQL.COLUMN_VALUE_LONG
- DBMS_SQL.COLUMN_VALUE_NUMBER
- DBMS_SQL.COLUMN_VALUE_RAW
- DBMS_SQL.COLUMN_VALUE_TIMESTAMP
- DBMS_SQL.COLUMN_VALUE_VARCHAR
- DBMS_SQL.DEFINE_COLUMN_BLOB
- DBMS_SQL.DEFINE_COLUMN_CHAR
- DBMS_SQL.DEFINE_COLUMN_CLOB
- DBMS_SQL.DEFINE_COLUMN_DATE
- DBMS_SQL.DEFINE_COLUMN_DOUBLE
- DBMS_SQL.DEFINE_COLUMN_INT
- DBMS_SQL.DEFINE_COLUMN_LONG
- DBMS_SQL.DEFINE_COLUMN_NUMBER
- DBMS_SQL.DEFINE_COLUMN_RAW
- DBMS_SQL.DEFINE_COLUMN_TIMESTAMP
- DBMS_SQL.DEFINE_COLUMN_VARCHAR
- DBMS_SQL.DESCRIBE_COLUMNS
- DBMS_SQL.DESCRIBE_COLUMNS2
- DBMS_SQL.EXECUTE
- DBMS_SQL.EXECUTE_AND_FETCH
- DBMS_SQL.EXECUTE_ROWS
- DBMS_SQL.IS_OPEN
- DBMS_SQL.LAST_ROW_COUNT
- DBMS_SQL.OPEN_CURSOR
- DBMS_SQL.PARSE
- DBMS_SQL.VARIABLE_VALUE
- DBMS_SQL.VARIABLE_VALUE
- DBMS_SQL.VARIABLE_VALUE
- DBMS_SQL.VARIABLE_VALUE
- DBMS_SQL.VARIABLE_VALUE
- DBMS_SQL.VARIABLE_VALUE
- DBMS_SQL.VARIABLE_VALUE
- DBMS_SQL.VARIABLE_VALUE
- DBMS_SQL.VARIABLE_VALUE
- DBMS_SQL.VARIABLE_VALUE
- DBMS_UTIL.ANALYZE_DATABASE
- DBMS_UTIL.ANALYZE_PART_OBJECT
- DBMS_UTIL.ANALYZE_SCHEMA
- DBMS_UTIL.CANONICALIZE
- DBMS_UTIL.COMMA_TO_TABLE

- DBMS_UTIL.COMPILE_SCHEMA
- DBMS_UTIL.DB_VERSION
- DBMS_UTIL.EXEC_DDL_STATEMENT
- DBMS_UTIL.GET_CPU_TIME
- DBMS_UTIL.GET_DEPENDENCY
- DBMS_UTIL.GET_HASH_VALUE
- DBMS_UTIL.GET_TIME
- DBMS_UTIL.NAME_RESOLVE
- DBMS_UTIL.NAME_TOKENIZE
- DBMS_UTIL.TABLE_TO_COMMA
- DBMS_UTIL.VALIDATE
- UTL_DIR.CREATE_DIRECTORY
- UTL_DIR.CREATE_OR_REPLACE_DIRECTORY
- UTL_DIR.DROP_DIRECTORY
- UTL_DIR.GET_DIRECTORY_PATH
- UTL_FILE.FCLOSE
- UTL_FILE.FCLOSE_ALL
- UTL_FILE.FCOPY
- UTL_FILE.FFLUSH
- UTL_FILE.FOPEN
- UTL_FILE.FREMOVE
- UTL_FILE.FRENAME
- UTL_FILE.GET_LINE
- UTL_FILE.IS_OPEN
- UTL_FILE.NEW_LINE
- UTL_FILE.PUT
- UTL_FILE.PUT_LINE
- UTL_FILE.PUTF
- UTL_FILE.FILE_TYPE
- UTL_MAIL.SEND
- UTL_MAIL.SEND_ATTACH_RAW
- UTL_MAIL.SEND_ATTACH_VARCHAR2
- UTL_SMTP.CLOSE_DATA
- UTL_SMTP.COMMAND
- UTL_SMTP.COMMAND_REPLIES
- UTL_SMTP.DATA
- UTL_SMTP.EHLO
- UTL_SMTP.HELO
- UTL_SMTP.HELP
- UTL_SMTP.MAIL
- UTL_SMTP.NOOP
- UTL_SMTP.OPEN_CONNECTION (function)
- UTL_SMTP.OPEN_CONNECTION (procedure)
- UTL_SMTP.OPEN_DATA
- UTL_SMTP.QUIT

- UTL_SMTP.RCPT
- UTL_SMTP.RSET
- UTL_SMTP.VRFY
- UTL_SMTP.WRITE_DATA
- UTL_SMTP.WRITE_RAW_DATA

The following monitor routines have been added:

- EVMON_FORMAT_UE_TO_TABLES
- EVMON_FORMAT_UE_TO_XML
- MON_GET_ACTIVITY_DETAILS
- MON_GET_BUFFERPOOL
- MON_GET_CONNECTION
- MON_GET_CONNECTION_DETAILS
- MON_GET_CONTAINER
- MON_GET_EXTENT_MOVEMENT_STATUS
- MON_GET_INDEX
- MON_GET_PKG_CACHE_STMT
- MON_GET_SERVICE_SUBCLASS
- MON_GET_SERVICE_SUBCLASS_DETAILS
- MON_GET_TABLE
- MON_GET_TABLESPACE
- MON_GET_UNIT_OF_WORK
- MON_GET_UNIT_OF_WORK_DETAILS
- MON_GET_WORKLOAD
- MON_GET_WORKLOAD_DETAILS

The following snapshot routines and views have been added:

- SNAP_GET_TBSP_PART_V97
- SNAP_GET_STORAGE_PATHS_V97

The following workload management routines have been added:

- WLM_GET_SERVICE_CLASS_AGENTS_V97
- WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES_V97
- WLM_GET_SERVICE_SUBCLASS_STATS_V97
- WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES_V97
- WLM_GET_WORKLOAD_STATS_V97

The following table functions have been deprecated in Version 9.7:

- HEALTH_CONT_HI
- HEALTH_CONT_HI_HIS
- HEALTH_CONT_INFO
- HEALTH_DB_HI
- HEALTH_DB_HI_HIS
- HEALTH_DB_HIC
- HEALTH_DB_HIC_HIS
- HEALTH_DB_INFO

- HEALTH_DBM_HI
- HEALTH_DBM_HI_HIS
- HEALTH_DBM_INFO
- HEALTH_GET_ALERT_ACTION_CFG
- HEALTH_GET_ALERT_CFG
- HEALTH_GET_IND_DEFINITION
- HEALTH_HI_REC
- HEALTH_TBS_HI
- HEALTH_TBS_HI_HIS
- HEALTH_TBS_INFO
- SNAP_GET_STORAGE_PATHS
- SNAP_GET_TBSP_PART_V91
- WLM_GET_ACTIVITY_DETAILS
- WLM_GET_SERVICE_CLASS_AGENTS
- WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES
- WLM_GET_SERVICE_SUBCLASS_STATS
- WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES
- WLM_GET_WORKLOAD_STATS

User response

Review the list of the "Deprecated SQL administrative routines and their replacement routines or views" in *Administrative Routines and Views* to determine additional changes that might impact your applications and scripts.

For the list of the data dictionary-compatible views, see the "Data dictionary-compatible views" topic.

New SYSIBM functions override unqualified user-defined functions with the same name

If you use the default SQL path (or an SQL path that has SYSIBM before user schemas), and the schema has existing functions with the same names as new SYSIBM functions, the SYSIBM functions are used instead. This situation usually improves performance, but might cause unexpected behaviors.

Details

If a user-defined function or a user-defined procedure has the same name and signature as a new built-in function or SQL administrative routine, an unqualified reference to those functions or routines in a dynamic SQL statement executes the built-in function or SQL administrative routine instead of the user-defined one.

The default SQL path contains the schemas SYSIBM, SYSFUN, SYSPROC, and SYSIBMADM before the schema name which is the value of the USER special register. These system schemas are also usually included in the SQL path when it is explicitly set using the SET PATH statement or the FUNCPATH bind option. When function resolution and procedure resolution is performed, the built-in functions and SQL administrative routines in the SYSIBM, SYSFUN, SYSPROC, and SYSIBMADM schemas are encountered before user-defined functions and user-defined procedures.

This change does not affect static SQL in packages or SQL objects such as views, triggers, or SQL functions which continue to execute the user-defined function or procedure until an explicit bind of the package, or drop and create of the SQL object.

For a complete list of scalar functions added in this release, refer to “Scalar function support has been extended” on page 79.

User response

Rename the user-defined routine or fully qualify the name before you invoke it. Alternatively, place in the SQL path the schema in which the user-defined routine exists before the schema in which the built-in functions and SQL administrative routines exist. However, promoting the schema in the SQL path increases the resolution time for all built-in functions and SQL administrative routines because the system schemas are considered first.

Related reference

″SET PATH ″ in SQL Reference, Volume 2

Untyped NULL keyword specifications no longer resolve to identifier names

Starting in Version 9.7, you can specify an untyped NULL keyword anywhere an expression is allowed. The behavior of existing expressions with NULL identifiers that are not qualified and not delimited might resolve to a null value instead of an identifier name and produce different results.

Details

To provide additional flexibility when creating expressions, you can now specify untyped NULL keyword specifications anywhere an expression is allowed. As a result, references to the NULL keyword that are not qualified and not delimited resolve to the null value when SQL statements are compiled rather than resolving to an identifier name as it was the case in earlier releases. For example, if a database identifier is named NULL and is used in an SQL statement without being fully qualified or delimited, the identifier specification might resolve to the keyword NULL instead of the identifier reference.

Assuming you have the following table and data:

```
CREATE TABLE MY_TAB (NULL INT)
INSERT INTO MY_TAB VALUES (1)
```

When you issue the following statement:

```
SELECT NULL FROM MY_TAB
```

In previous releases, the null that is specified in the select list resolves to the column named NULL. Starting in Version 9.7, it resolves to the null value.

Also, when you issue the following statement:

```
SELECT NULL FROM TABLE(VALUES(1)) AS X(NULL)
```

In previous releases, this statement returns a value of 1. Starting in Version 9.7, this statement returns a null value.

User response

To avoid conflicts with the keyword NULL, columns named NULL should be fully qualified or delimited when used in SQL statements.

Review existing expressions that use untyped NULL keywords specifications, and update them if needed. You can use the db2ckupgrade command to check for identifiers named “NULL”.

For example, the following expressions can be used to produce the behaviors from previous releases:

```
SELECT MY_TAB.NULL FROM MY_TAB
SELECT "NULL" FROM MY_TAB
```

Related reference

″db2ckupgrade - Check database for upgrade ″ in Command Reference

CHAR(*decimal-expression*) scalar function return behavior has been changed

In databases created in Version 9.7, leading zeros and a trailing decimal character are removed from the result of the CHAR scalar function (decimal to character). This behavior also applies to the CAST specification from decimal to character.

Details

In previous releases, the CHAR scalar function (decimal to character) as well as the CAST specification from decimal to character returns leading zeros and a trailing decimal character in the result. The behavior is inconsistent with the VARCHAR scalar function as well as the SQL standard casting rules.

The following examples illustrate the changed behavior:

Example 1

Assuming that you create the following table and data:

```
CREATE TABLE MY_TAB (C1 DEC(31,2)
INSERT INTO MY_TAB VALUES 0.20, 0.02, 1.20, 333.44
```

When you issue the following statement:

```
SELECT CHAR(C1)FROM MY_TAB
```

In previous releases, the following result set is returned:

```
1
---------------------------------
00000000000000000000000000000.20
00000000000000000000000000000.02
00000000000000000000000000001.20
00000000000000000000000000333.44
```

In Version 9.7, the following result set is returned:

```
1
---------------------------------
.20
.02
1.20
333.44
```

Example 2

Assuming that you create the following table and data:

```
CREATE TABLE MY_TAB (C1 DEC(5,0))
INSERT INTO MY_TAB VALUES 1, 4.0
SELECT CHAR(C1)FROM MY_TAB
```

When you issue the following statement:

```
SELECT CHAR(C1)FROM MY_TAB
```

In previous releases, the following result set is returned:

```
1
-----
0001.
0004.
```

In Version 9.7, the following result set is returned:

```
1
-----
1
4
```

User response

If previous release semantics are required, you can use the following methods:

- Update the specific SQL statements to use the CHAR_OLD(<decimal>) scalar function rather than the CHAR(<decimal>) scalar function.
- Update the database configuration parameter **dec_to_char_fmt** to 'V95'. After setting the database configuration parameter, SQL statements using the CHAR scalar function or the CAST specification from decimal to character will need to be recompiled. For static SQL, you must rebind the package. For dynamic SQL, the statement simply needs to be invoked.

If you want migrated databases to use the new format, set the **dec_to_char_fmt** to 'NEW'.

Related reference

"CHAR " in SQL Reference, Volume 1

"dec_to_char_fmt - Decimal to character function configuration parameter" in Database Administration Concepts and Configuration Reference

DOUBLE(*string-expression*) scalar function return behavior has been changed

In Version 9.7, leading and trailing blanks are removed from the argument *string-expression* of the DOUBLE(character string to double) scalar function. If the resulting *string-expression* argument is an empty string, an error is returned rather than returning the +0.00000000000000E+000 value.

Details

In previous releases, the DOUBLE scalar function (character string to double) removes leading and trailing whitespace (blanks, tabs, carriage-return, newline, vertical tab and form-feed) from the *string-expression* before converting the argument to a floating-point number. This behavior is inconsistent with the documentation for the scalar function, other numeric scalar functions, and other database products in the DB2 family.

In Version 9.7, the support for the DOUBLE scalar function has been extended to the SYSIBM schema, making it a built-in function as well, and the handling of leading and trailing whitespace has been changed. As a result, an error (SQLSTATE 22018) is returned in the following situations:

- *string-expression* contains whitespace characters other than a blank
- *string-expression* contains only blanks
- *string-expression* is an empty string

User response

If the previous releases semantics are required, you can use the SYSFUN version of the DOUBLE scalar function using any of the following methods:

- You can fully qualify the reference to the scalar function with SYSFUN. For example, SYSFUN.DOUBLE(*string-expression*).
- You can create a sourced function on SYSFUN.DOUBLE and include the schema of the function before SYSIBM in the SQL path.
- You can place the SYSFUN schema in the SQL path before the SYSIBM schema. However, this is not recommended because it will affect many other functions as well.

Related reference

"DOUBLE_PRECISION or DOUBLE" in SQL Reference, Volume 1

Result data type for unary minus and unary operators in untyped expressions has been changed

Starting with Version 9.7, unary minus or unary plus operators in untyped expressions return DECFLOAT(34).

Details

In previous releases, the result data type of a unary minus or unary plus operators with an untyped expression as an argument is DOUBLE.

User response

If previous release semantics are required, you can cast the untyped expression explicitly to DOUBLE. For example:

```
–(CAST (? AS DOUBLE))
```

DEFAULT keyword specification has been changed

Starting in Version 9.7, an unqualified and non-delimited reference to DEFAULT always resolves to the DEFAULT keyword. As a result, the behavior of procedures that use DEFAULT as parameters and the behavior of some SQL PL assignment statements have been changed.

Details

Using non-delimited references to DEFAULT on the right side of an SQL PL assignment statement no longer resolves to a variable or parameter named DEFAULT. Instead, it resolves to the DEFAULT keyword. If the usage of the DEFAULT keyword is invalid, an error is returned (SQLSTATE 42608).

In addition, the invocation of a procedure which specifies DEFAULT as a parameter always resolve to the DEFAULT keyword, rather than resolving to a variable or parameter named DEFAULT if the variable or parameter exists. This change enables you to specify DEFAULT as a parameter value for procedure invocation.

In previous releases, SQL PL assignment statements in the form `"SET V = DEFAULT"` , where *V* is a local variable, produces one of the following results:

- DEFAULT resolves to a variable or parameter, if one is defined
- An error (SQLSTATE 42608) is returned, if a variable or parameter with the name DEFAULT is not defined

This behavior is inconsistent with the assignment to global variables as well as the VALUES statement, where specifying DEFAULT always resolves to the DEFAULT keyword.

Also in previous releases, the invocation of a procedure specifying DEFAULT as a parameter produces one of the following results:

- The variable or parameter resolves to a variable or parameter named DEFAULT, if one is defined.
- An error (SQLSTATE 42608) is returned, if a variable or parameter named DEFAULT is not defined

User response

To avoid conflicts with the keyword DEFAULT, you should delimit variables named DEFAULT using double quotes and use uppercase letters when using these variables in SQL and SQL PL assignment statements, as well as in procedure invocations.

Example

When you create and call the following procedure:

```
CREATE PROCEDURE foo(IN DEFAULT INTEGER)
BEGIN
   DECLARE V0 INTEGER DEFAULT 1;
   SET V0 = "DEFAULT";
   RETURN V0;

END%

CALL foo(10)%
```

It returns correctly the following data:

```
Return Status = 10
```

XML data is passed by reference in SQL stored procedures

In SQL stored procedures, when you assign XML data to XML input, output, or input/output parameters or XML local variables, the XML values are now passed by reference. Therefore, some operations using XML data return results that are different from the results returned by the same operations in DB2 Version 9.5 and earlier.

Details

When you assign XML data to a parameter or local variable and the values are passed by reference, the node identities and the parent property are preserved. Therefore, results of the following types of operations might change:

- Operations that use the node identities of the XML value
- Operations that use the parent axis in an XPath expression

The following types of expressions use node identity:
- Node comparisons. The IS operator uses node identity to determine if two nodes have the same identity. The >> operator and << operator use node identity to compare the document order of nodes.
- Path expressions. Path expressions use node identity to eliminate duplicate nodes.
- Sequence expressions. The UNION, INTERSECT, and EXCEPT operators use node identity to eliminate duplicate nodes.

In DB2 Version 9.5 and earlier, when you assign XML data to a parameter or local variable, the XML data is passed by value. Therefore, node identities and the parent property are not preserved.

User response

Check the stored procedure to ensure that it returns the correct results when performing operations comparing node identities and operations that use the parent axis in path expressions.

Example

The stored procedure in the example shows that different results are returned when passing XML data by reference and by value.

The stored procedure uses data from a table that contains an XML column and returns results into a second table. The following statements create the tables and insert data into the first table:

```
CREATE TABLE t1 (c1 INT, c2 XML) ~
INSERT INTO t1 VALUES (1, '<a><b><d>1</d></b><c>2</c></a>') ~

CREATE TABLE t2 (c1 INT, c2 VARCHAR(1000)) ~
 ~
```

The stored procedure assigns the XML data from the XML column to two XML variables and performs operations that return different results depending on the version of the DB2 database server being used:

```
CREATE PROCEDURE MYTESTPROC (  )
BEGIN
  DECLARE v1, v2, v3 XML;

  -- Assign XML value to v1 and v2
  SELECT XMLQUERY('$c/a/b' passing by ref c2 as "c") INTO v1
  FROM t1  WHERE c1 = 1;

  SELECT XMLQUERY('$c/a/b' passing by ref c2 as "c") INTO v2
  FROM t1  WHERE c1 = 1;

  -- insert XML value into t2
  INSERT INTO t2 VALUES (1, xmlserialize(v1 as VARCHAR(1000)));

  -- OR operator combining sequences of nodes
  -- If node identities are identical, sequence expression will drop duplicate nodes
  SET v3 = xmlquery ('$x | $y' passing v1 as "x", v2 as "y");
  INSERT INTO t2 VALUES (2, xmlserialize(v3 as VARCHAR(1000)));

  -- Creating a sequnce of nodes
  SET v3 = xmlquery ('($x,$y)' passing v1 as "x", v2 as "y");
  -- If node identities are identical, path expression will drop duplicate nodes
  SET v3 = xmlquery ('$x/d' passing v3 as "x");
  INSERT INTO t2 VALUES (3, xmlserialize(v3 as VARCHAR(1000)));

  -- Test of parent axis property
  SET v3 = xmlquery('$x/..' passing v1 as "x");
  INSERT INTO t2 VALUES (4, xmlserialize(v3 as VARCHAR(1000)));

  -- NODE ID comparison
  if(xmlcast(xmlquery('$X is $Y' passing by ref v1 as X, v2 as Y) as VARCHAR(5))='true') then
    INSERT INTO t2 VALUES (5, 'NODE ID preserved');
  else
   INSERT INTO t2 VALUES (5, 'NODE ID NOT preserved');
  end if;
END
```

The stored procedure returns the following values for the different versions of the DB2 database server.

Table 26. Stored procedure values inserted to table t2

Column c1	DB2 Version 9.7 (passing by reference)	DB2 Version 9.5 (passing by value)
1	<b><d>1</d></b>	<b><d>1</d></b>
2	<b><d>1</d></b>	<b><d>1</d></b><b><d>1</d></b>
3	<d>1</d>	<d>1</d><d>1</d>
4	<a><b><d>1</d></b><c>2</c></a>	NULL
5	NODE ID preserved	NODE ID NOT preserved

Related information

″Axes″ in XQuery Reference

″Node comparisons″ in XQuery Reference

″Node identity″ in pureXML Guide

″Expressions for combining sequences of nodes″ in XQuery Reference

Type annotations for validated XML documents are unavailable

In Version 9.7, validated XML documents are not augmented with type annotations. Validated XML documents from Version 9.5 or earlier have type annotations, but Version 9.7 does not use them. Type information is stripped from element and attribute nodes that are copied to form the content of a newly constructed node.

Details

The XMLVALIDATE function now annotates a successfully validated XML document only with information about the schema used to validate the document. The function does not augment the element and attribute nodes with type information. Element node values or attribute values from validated documents returned in XQuery expressions are represented using a string data type. If the data is defined in the schema as xs:list, it is represented as xdt:untypedAtomic.

The output of the VALIDATED predicate and the XMLXSROBJECTID function remain the same. The VALIDATED predicate tests whether or not an XML document has been validated using the XMLVALIDATE function, and the XMLXSROBJECTID function returns the XSR object identifier of the XML schema used to validate an XML document.

In the DB2 XQuery prolog, the default XML construction declaration value has been changed from preserve to strip. The construction declaration value sets the construction mode for the query. When the construction mode is strip, type information is stripped from element and attribute nodes that are copied to form the content of a newly constructed node.

User response

For existing applications that use XQuery and validated XML documents, modify the XQuery expressions to cast data to the appropriate type to ensure correct results.

When creating new applications using DB2 pureXML, be aware that all XQuery comparisons are string comparisons unless you cast the data to a different data type. For example, without casting, XQuery operators such as the greater than (>) and less than (<) operators compare node and attribute values as strings, and the XQuery ORDER BY clause sorts data as strings.

To process data that is defined in the XML schema as xs:list as a list, use the fn:tokenize function to convert it to a sequence.

Creating indexes over XML data

The successful creation of an index over XML data depends on the compatibility of the XML values as type xdt:untypedAtomic to the SQL type specified for the index. If an XML value is not compatible with the SQL type during index creation, the error message SQL20306N is returned with error code 4. In DB2 Version 9.5 or earlier, either error code 2 or 3 was returned. If an XML value is not compatible with the SQL type specified for an index over XML data when you insert or update XML documents, the error message SQL20305N is returned with error code 4. In DB2 Version 9.5 and earlier, either error code 2 or 3 was returned.

Matching indexes over XML data

Type casting is required to match indexes over XML data that specifies only DOUBLE and DATETIME data types. Indexes over XML data that specify the VARCHAR data type are implied to match a query against the XML data if there is no type casting. You do not need to use the functions fn:string or xs:string to convert data from validated XML documents for matching indexes over XML data.

Related concepts

″XML validation″ in pureXML Guide

Related reference

″XMLVALIDATE ″ in SQL Reference, Volume 1

Merge modules for ODBC, CLI, and .NET have been combined (Windows)

IBM Data Server Driver for ODBC, CLI, and .NET has been renamed to IBM Data Server Driver Package, which continues to provide an MSI-based Windows installation that uses merge modules. However, the packaging strategy has been simplified in Version 9.7 to provide a single merge module for ODBC, CLI, and .NET rather than multiple merge modules.

Details

The contents of the old IBM Data Server Driver for ODBC and CLI Merge Module.msm and IBM Data Server Provider for .NET Merge Module.msm merge modules are now available in a single merge module, named the IBM Data Server Driver Package.msm merge module. The old merge modules for ODBC, CLI, and .NET are no longer available.

This change does not affect the language-specific merge modules, which continue to be available separately.

User response

Update references to the ODBC and CLI merge module and references to the .NET merge module to use the new merge module name.

Related concepts

“Component names have changed” on page 3

Related reference

″Non-DB2 instance merge modules (Windows)″ in Installing IBM Data Server Clients

Result data type for integer division in number_compat mode has been changed

Starting with Version 9.7, when a database has been created in number_compat mode, the result data type of division operations involving only integer expressions returns DECFLOAT(34) and the operation is performed using decimal floating point arithmetic. This result for integer division is consistent with results in compatible databases that support the NUMBER data type.

Details

In the previous release, when you created a database with the **DB2_COMPATIBILITY_VECTOR** registry variable set to enable the NUMBER data type, the result data type of an integer division was an integer data type and the operation was performed using binary integer arithmetic.

An upgraded database might include SQL objects with expressions that are impacted by this change. The result type for view columns involving integer division might change. If an expression involving integer division is used as the argument of a function, the result of function resolution might be different.

User response

In most cases, the implicit casting support included in Version 9.7 will implicitly handle the change in the data type of the expression. If the usage of an SQL object fails because of the change in data type, extract the object definition statement from the catalog or use db2look, change the CREATE option to the CREATE OR REPLACE option in the statement and run the statement again. This will replace the object in the upgraded database using the new result data type for division operations involving integer expressions.

Related reference

″DB2_COMPATIBILITY_VECTOR registry variable″ in SQL Procedural Languages: Application Enablement and Support

″NUMBER data type″ in SQL Procedural Languages: Application Enablement and Support

Chapter 17. Deprecated functionality

Functionality gets marked as *deprecated* when a specific function or feature is supported in the current release but might be removed in a future release. In some cases, it might be advisable to plan to discontinue the use of deprecated functionality.

For example, a registry variable might be deprecated in this release because the behavior triggered by the registry variable has been enabled by default in this release, and the obsolete registry variable will be removed in a future release.

The following DB2 components and associated functionality have been deprecated:
- Control Center tools and DB2 administration server (see "Control Center tools and DB2 administration server (DAS) have been deprecated" on page 174)
- DB2 Governor and Query Patroller (see "DB2 Governor and Query Patroller have been deprecated" on page 175)
- Health monitor (see "Health monitor has been deprecated" on page 177)

Additionally, the following functionality about general DB2 support has been deprecated:
- LONG VARCHAR and LONG VARGRAPHIC data types (see "LONG VARCHAR and LONG VARGRAPHIC data types have been deprecated" on page 178)
- Worksheet Format (WSF) for Export and Load utilities (see "Worksheet Format (WSF) for Export and Load utilities has been deprecated" on page 178)
- LIST TABLESPACES and LIST TABLESPACE CONTAINERS commands and related APIs (see "LIST TABLESPACES and LIST TABLESPACE CONTAINERS commands have been deprecated" on page 179)
- SDK 1.4.2 support for Java routines (see "IBM Software Developer's Kit (SDK) 1.4.2 support for Java routines has been deprecated" on page 179)
- sqlugrpn API (see "sqlugrpn API has been deprecated" on page 180)
- sqlugtpi API (see "sqlugtpi API has been deprecated" on page 180)
- Subset of Net Search Extender features and commands (see "Subset of Net Search Extender features and commands have been deprecated" on page 181)
- Functionality related to the discontinued type-1 indexes (see "Type-1 indexes have been discontinued" on page 187)
- **DB2_CAPTURE_LOCKTIMEOUT** and the **DB2_SERVER_ENCALG** registry variables (see "Some registry and environment variables have been deprecated" on page 183)

The following monitoring functionality has been deprecated:
- CREATE EVENT MONITOR FOR DEADLOCKS statement as well as the DB2DETAILDEADLOCK event monitor (see "CREATE EVENT MONITOR FOR DEADLOCKS statement and DB2DETAILDEADLOCK event monitor have been deprecated" on page 182)
- CREATE EVENT MONITOR FOR TRANSACTIONS statement (see "CREATE EVENT MONITOR FOR TRANSACTIONS statement has been deprecated" on page 182)

The following product installation and instance management functionality has been deprecated:

- **-s** option of the db2iupdt command on Linux and UNIX operating systems (see "db2iupdt command -s option has been deprecated (Linux and UNIX)" on page 183)
- Instance and database migration commands and APIs (see "Instance and database migration commands and APIs have been deprecated" on page 184)
- MIGRATE_PRIOR_VERSIONS and CONFIG_ONLY response file keywords (see "Some response file keywords have been deprecated" on page 185)

Review each topic to find out more details and to plan for future changes. Other deprecated functionality might be described in Chapter 18, "Discontinued functionality," on page 187.

Control Center tools and DB2 administration server (DAS) have been deprecated

Starting in Version 9.7, the Control Center tools and the DAS have been deprecated and might be removed in a future release. A new suite of GUI tools for managing DB2 for Linux, UNIX, and Windows data and data-centric applications is now available and can be used instead.

Details

The following Control Center tools and related features have been deprecated:

- Activity Monitor
- Command Editor
- Configuration Assistant
- Control Center and associated wizards and advisors
- Control Center plug-in extensions
- DB2 administration server (DAS)
- Event Analyzer
- Health Center
- Indoubt Transaction Monitor
- Journal
- License Center
- Memory Visualizer
- Query Patroller Center
- Replication Center
- Satellite Administration Center
- Task Center

As a result, the following associated DB2 commands are also deprecated:

- dasauto (Autostart DB2 administration server command)
- dascrt (Create a DB2 administration server command)
- dasdrop (Remove a DB2 administration server command)
- dasmigr (Migrate the DB2 administration server command)
- dasupdt (Update DAS command)
- daslist (Display DAS name command)

- db2admin (DB2 administration server command)
- db2am (Start Activity monitor center command)
- db2ca (Start the Configuration Assistant command)
- db2cc (Start control center command)
- db2ce (Start Command Editor command)
- db2daslevel (Show DAS level command)
- db2eva (Event analyzer command)
- db2hc (Start health center command)
- db2indbt (Start Indoubt Transaction Monitor Center command)
- db2journal (Start Journal command)
- db2lc (Start License Center command)
- db2memvis (Start Memory Visualizer center command)
- db2rc (Start Replication Center command)
- db2tc (Start Task Center command)

User response

The deprecated Control Center tools and related features are still supported in Version 9.7. However, you should consider using the new suite of GUI tools in place of the Control Center tools. For more information, see Database management and application development tools.

DB2 Governor and Query Patroller have been deprecated

Due to the introduction of DB2 workload manager as the strategic workload management solution in DB2 Version 9.5, Query Patroller and DB2 Governor have been deprecated and might be removed in a future release.

Details

Together, Query Patroller and DB2 Governor provide workload management controls to successfully run complex workloads on your DB2 data server. However, DB2 workload manager provides a greatly enhanced set of workload management features that replaces both Query Patroller and DB2 Governor.

All Query Patroller components have been deprecated, including:

- The Query Patroller server (including Query Patroller stored procedures, control tables, and log files)
- Query Patroller Center
- Query Patroller thresholds
- Query Patroller historical analysis functionality
- Query Patroller registry variables: **DB2_QP_BYPASS_APPLICATIONS**, **DB2_QP_BYPASS_USERS**, **DB2_QP_BYPASS_COST**
- The qp_query_id monitor element
- The **dyn_query_mgmt** database configuration parameter
- Query Patroller command-line support, including the following commands:
  - ADD OPERATOR_PROFILE
  - ADD QUERY_CLASS
  - ADD SUBMISSION_PREFERENCES
  - ADD SUBMITTER_PROFILE

  - CANCEL QUERY
  - GENERATE HISTORICAL_DATAFILE RESULT
  - GET OPERATOR_PROFILE
  - GET QP_SYSTEM
  - GET QUERY
  - GET QUERY_CLASS
  - GET SUBMISSION_PREFERENCES
  - GET SUBMITTER_PROFILE
  - LIST OPERATOR_PROFILES
  - LIST QUERIES
  - LIST QUERY_CLASSES
  - LIST SUBMISSION_PREFERENCES
  - LIST SUBMITTER_PROFILES
  - qpcenter
  - qpsetup
  - qpstart
  - qpstop
  - REMOVE OPERATOR_PROFILE
  - REMOVE QUERY_CLASS
  - REMOVE QUERY_INFO
  - REMOVE QUERY_INFO_HISTORY
  - REMOVE RESULT
  - REMOVE RESULT_TABLE_ALIASES
  - REMOVE SUBMISSION_PREFERENCES
  - REMOVE SUBMITTER_PROFILE
  - RUN HELD_QUERY
  - RUN IN BACKGROUND QUERY
  - SHOW RESULT
  - UPDATE OPERATOR_PROFILE
  - UPDATE QUERY_CLASS
  - UPDATE SUBMISSION_PREFERENCES
  - UPDATE SUBMITTER_PROFILE
  - UPDATE QP_SYSTEM

In addition, all DB2 Governor commands have been deprecated, including:

- db2gov
- db2govlg

User response

You can manage workloads more effectively by using DB2 workload manager, which provides many more features.

Related concepts

″Workload management roadmap″ in Workload Manager Guide and Reference

″Frequently asked questions about DB2 workload manager″ in Workload Manager Guide and Reference

"New thresholds provide additional activity control" on page 58

Health monitor has been deprecated

The health monitor has been deprecated. A new suite of GUI tools for managing DB2 for Linux, UNIX, and Windows data and data-centric applications is now available and can be used instead.

Details

The following APIs, API options and value have been deprecated in Version 9.7:

- db2GetAlertCfg
- db2GetAlertCfgFree
- db2GetRecommendations
- db2GetRecommendationsFree
- db2ResetAlertCfg
- db2UpdateAlertCfg
- The **SQLM_CLASS_HEALTH** and **SQLM_CLASS_HEALTH_WITH_DETAIL** snapshot class options of the db2GetSnapshot API
- The SQLM_HMON_OPT_COLL_FULL value for the **AGENT_ID** option in the sqlma data structure passed to the db2GetSnapshot API

The following CLP commands have been deprecated:

- GET ALERT CONFIGURATION
- GET HEALTH SNAPSHOT
- GET RECOMMENDATIONS FOR HEALTH INDICATOR
- RESET ALERT CONFIGURATION
- UPDATE ALERT CONFIGURATION

The following table functions have been deprecated:

- HEALTH_CONT_HI
- HEALTH_CONT_HI_HIS
- HEALTH_CONT_INFO
- HEALTH_DB_HI
- HEALTH_DB_HI_HIS
- HEALTH_DB_HIC
- HEALTH_DB_HIC_HIS
- HEALTH_DB_INFO
- HEALTH_DBM_HI
- HEALTH_DBM_HI_HIS
- HEALTH_DBM_INFO
- HEALTH_GET_ALERT_ACTION_CFG
- HEALTH_GET_ALERT_CFG
- HEALTH_GET_IND_DEFINITION

- HEALTH_HI_REC
- HEALTH_TBS_HI
- HEALTH_TBS_HI_HIS
- HEALTH_TBS_INFO

User response

The deprecated health monitor interfaces are still supported in Version 9.7. However, you should consider using the new suite of GUI tools in place of the Control Center tools. For more information, see Database management and application development tools.

LONG VARCHAR and LONG VARGRAPHIC data types have been deprecated

The LONG VARCHAR and LONG VARGRAPHIC data types have been deprecated. As a result, the LONG_VARGRAPHIC and LONG_VARCHAR scalar functions have also been deprecated.

Details

When choosing a data type for a column, use data types such as VARCHAR, VARGRAPHIC, CLOB, or DBCLOB since these will continue to be supported in future releases and are recommended for portable applications.

User response

The usage of LONG VARCHAR and LONG VARGRAPHIC does not affect existing tables because deprecated functionality continues to be supported in the current release. Consider migrating to other data types to ensure that you can take advantage of future enhancements to the product. The support for the LONG VARCHAR and LONG VARGRAPHIC data types and the associated scalar functions might be removed in a future release.

Also, in embedded SQL applications, avoid using host variables that generate the deprecated data types.

Worksheet Format (WSF) for Export and Load utilities has been deprecated

The Worksheet Format (WSF) was used for data exchange with products such as Lotus® 1-2-3® and Symphony. Support for the file format is deprecated and might be removed in a future release.

Details

WSF files have limitations relative to other supported file formats. This format is not recommended for DB2 utilities.

User response

Start using a supported file format instead of WSF files before support is removed.

For existing WSF files, convert to another format by loading the data back into DB2 tables and exporting the data to a supported format such as ASC, DEL, or PC/IXF.

LIST TABLESPACES and LIST TABLESPACE CONTAINERS commands have been deprecated

The commands and APIs that show information about table spaces and table space containers have been deprecated and might be removed in a future release.

Details

The LIST TABLESPACES [SHOW DETAIL] and LIST TABLESPACE CONTAINERS commands are no longer being updated with new features.

As a result, the following data structure and APIs are also deprecated:

- SQLB_TBSPQRY_DATA data structure
- sqlbctsq API
- sqlbftsq API
- sqlbftpq API
- sqlbgtss API
- sqlbmtsq API
- sqlbotsq API
- sqlbstpq API
- sqlbstsq API
- sqlbtcq API

User response

Modify any of your existing scripts using the deprecated commands or APIs to call the MON_GET_TABLESPACE or the MON_GET_CONTAINER table functions instead. These table functions return more information than was provided by the deprecated commands and APIs.

IBM Software Developer's Kit (SDK) 1.4.2 support for Java routines has been deprecated

IBM SDK Version 1.4.2 support for Java routines has been deprecated. As a result, Java stored procedures and routines built in DB2 Version 8.2 (and earlier) are also deprecated because they were created using the SDK 1.4.2 (and earlier) level.

Details

Due to the introduction of newer versions of the IBM SDK, support for SDK Version 1.4.2 is deprecated and will be removed from service.

DB2 Version 9.7 installs IBM SDK for Java 6 by default on all platforms. This version of Java will be used to compile new Java stored procedures and user-defined functions created in Version 9.7.

User response

Before IBM SDK 1.4.2 support is discontinued, recreate any deprecated Java routines with the SDK installed in your DB2 Version 9.7 copy.

If you must use an SDK for Java other than the one installed in your DB2 Version 9.7 copy, refer to the refer to the “Upgrading Java routines” topic. For the list of supported Java development software, see “Java software support for DB2 products”.

Related reference

″Java software support for DB2 products″ in Installing and Configuring DB2 Connect Servers

sqlugrpn API has been deprecated

The sqlugrpn API, which retrieves the distribution map offset and database partition numbers for a row, has been deprecated and might be removed in a future release.

Details

The sqlugrpn API is designed to work with distribution maps containing up to 4096 (4 KB) entries.

In Version 9.7, the distribution map size has been expanded to 32 768 (32 KB) entries. The sqlugrpn API cannot be used in conjunction with these larger distribution maps. For information about enabling larger maps, see the “Distribution maps” topic.

User response

Use the new db2GetRowPartNum API, which supports all distribution map sizes.

Related concepts

″Distribution maps″ in Partitioning and Clustering Guide

“sqlugtpi API has been deprecated”

sqlugtpi API has been deprecated

The sqlugtpi API, which gets table distribution information, has been deprecated and might be removed in a future release.

Details

The sqlugtpi API is designed to work with distribution maps containing up to 4096 (4 KB) entries.

In Version 9.7, the distribution map size has been expanded to 32 768 (32 KB) entries. The sqlugtpi API cannot be used in conjunction with larger distribution maps that make use of more than 4096 entries. For information about enabling larger maps, see the “Distribution maps” topic.

User response

If the sqlugtpi API encounters a distribution map that it cannot process due to the larger size, it returns SQL2768N. Use the DB2GetDistMap API, which supports all distribution map sizes.

Related concepts

″Distribution maps″ in Partitioning and Clustering Guide

"sqlugrpn API has been deprecated" on page 180

Related reference

″db2GetDistMap - Get distribution map″ in Administrative API Reference

Subset of Net Search Extender features and commands have been deprecated

Certain Net Search Extender (NSE) features and commands have been deprecated and might be removed in a future release.

Details

The following NSE features have been deprecated:

- Index updates using replication
- Indexes on nicknames (federated databases)
- The NUMBEROFMATCHES scalar function
- Highlighting in search operations
- User-defined relationships in a thesaurus
- Documents in General Purpose Format (GPP)
- Caching for search operations using the stored procedures
- Presorted indexes used in search operations using the stored procedures

The following commands have been deprecated because the related features have also been deprecated:

- ACTIVATE CACHE
- DEACTIVATE CACHE
- DB2EXTHL

User response

You can start using other supported features or commands before these deprecated features and commands become discontinued. Avoid using deprecated features and commands when developing new applications.

Related concepts

″Net Search Extender k ey concepts″ in Net Search Extender Administration and User's Guide

CREATE EVENT MONITOR FOR DEADLOCKS statement and DB2DETAILDEADLOCK event monitor have been deprecated

The use of the CREATE EVENT MONITOR FOR DEADLOCKS statement and the automatically started DB2DETAILDEADLOCK event monitor, to monitor deadlock events, have been deprecated. Their use is no longer recommended and might be removed in a future release.

Details

In previous releases, if you wanted to monitor deadlock events, you had to issue the CREATE EVENT MONITOR FOR DEADLOCKS statement or check the output files for deadlock-related entries written by the automatically started DB2DETAILDEADLOCK event monitor. Version 9.7 includes a new event monitor infrastructure that provides an entirely new set of monitor elements and methods to monitor DB2 events. As a result, if you want to monitor deadlock events in DB2 Version 9.7, using the CREATE EVENT MONITOR FOR LOCKING statement is the suggested method.

User response

Use the CREATE EVENT MONITOR FOR LOCKING statement to monitor lock-related events, such as lock timeouts, lock waits, and deadlocks.

Related concepts

"New relational monitoring interfaces are light weight and SQL accessible" on page 29

Related reference

″CREATE EVENT MONITOR (locking) ″ in SQL Reference, Volume 2

CREATE EVENT MONITOR FOR TRANSACTIONS statement has been deprecated

The use of the CREATE EVENT MONITOR FOR TRANSACTIONS statement to monitor transaction events has been deprecated. Its use is no longer recommended and might be removed in a future release.

Details

In previous releases, if you wanted to monitor transaction events, you had to issue the CREATE EVENT MONITOR FOR TRANSACTIONS statement to create a transaction event monitor. Version 9.7 includes a new event monitor infrastructure that provides an entirely new set of monitor elements and methods to monitor DB2 events. As a result, if you want to monitor transaction events in DB2 Version 9.7, using the CREATE EVENT MONITOR FOR UNIT OF WORK statement is the suggested method.

User response

Use the CREATE EVENT MONITOR FOR UNIT OF WORK statement to create a transaction event monitor.

Related reference

″CREATE EVENT MONITOR (unit of work) ″ in SQL Reference, Volume 2

Some registry and environment variables have been deprecated

The **DB2_CAPTURE_LOCKTIMEOUT** and **DB2_SERVER_ENCALG** registry variables have been deprecated in Version 9.7. The variables are still available, but you should not use them because they will likely be removed in a future version.

The following table lists deprecated registry and environment variables. They have been replaced by another feature, or the function that they support is obsolete.

Table 27. Registry and environment variables deprecated in Version 9.7

Registry or environment variable	Details
DB2_CAPTURE_LOCKTIMEOUT	The registry variable is being deprecated and might be removed in a future release because there are new methods to collect lock timeout events using the CREATE EVENT MONITOR FOR LOCKING statement. For more information, see “Lock event reporting has been enhanced” on page 34.
DB2_SERVER_ENCALG	The registry variable is being deprecated and might be removed in a future release because you should use the **alternate_auth_enc** configuration parameter instead. For more information, see “AES encryption of user ID and password enhances security” on page 66.

Related concepts

“Some registry and environment variables have changed” on page 136

db2iupdt command -s option has been deprecated (Linux and UNIX)

The **-s** option of the db2iupdt command has been deprecated and might be removed in a future release.

Details

The db2iupdt command updates an instance to run on a DB2 copy that has a new DB2 database product or feature installed, to run on a DB2 copy of the same version as the DB2 copy associated with the instance, or to update the instance type to a higher level instance type. On UNIX and Linux operating systems, the **-s** parameter ignores the existing sync point manager (SPM) log directory.

User response

On UNIX and Linux operating systems, do not use this option of the db2iupdt command.

Instance and database migration commands and APIs have been deprecated

The db2imigr, db2ckmig, and MIGRATE DATABASE commands as well as the sqlemgdb and sqlgmgdb APIs have been deprecated in DB2 Version 9.7 and might be removed in a future release.

Details

To be consistent with the use of the term *upgrade* for DB2 products, the term upgrade is now used to describe the process of enabling pre-Version 9.7 DB2 servers, clients, database applications, and routines to run in a Version 9.7 environment. Pre-Version 9.7 refers only to DB2 Universal Database Version 8, DB2 Version 9.1, and DB2 Version 9.5.

The term upgrade is also used to describe the process of enabling pre-Version 9.7 instances and databases to run in a DB2 Version 9.7 copy.

Prior to Version 9.7, the term *migration* was used to describe the process of enabling DB2 servers, clients, database applications, routines, instances, and databases from one release to run in a later release.

Because of this change in terminology, the DB2 commands and APIs to migrate instances and databases have been deprecated, and new commands and APIs to upgrade instances and databases are available. The following table shows new Version 9.7 commands and APIs that you should use.

Table 28. Equivalent pre-Version 9.7 and Version 9.7 commands

Pre-Version 9.7 command or API name	**Version 9.7 command or API name**	**Version 9.7 command or API description**
db2imigr	db2iupgrade	db2iupgrade upgrades an instance to Version 9.7 from Version 8, Version 9.1, or Version 9.5.
db2ckmig	db2ckupgrade	db2ckupgrade verifies that the local pre-Version 9.7 databases are ready to be upgraded to Version 9.7.
MIGRATE DATABASE	UPGRADE DATABASE	UPGRADE DATABASE upgrades a database to Version 9.7 if the instance where the database was running was upgraded to Version 9.7 by using the db2iupgrade command.
sqlemgdb and sqlgmgdb	db2DatabaseUpgrade	db2DatabaseUpgrade converts a DB2 Version 9.5, Version 9.1, or Version 8 database to the current release.

User response

Use the new DB2 commands and APIs provided in Version 9.7 when upgrading instances and databases to Version 9.7.

Refer to ″Upgrade to DB2 Version 9.7″ for details about the complete upgrade process for DB2 servers, clients, database applications, and routines.

Related concepts

″Upgrade to DB2 Version 9.7″ in Upgrading to DB2 Version 9.7

Related reference

″UPGRADE DATABASE ″ in Command Reference

″db2ckupgrade - Check database for upgrade ″ in Command Reference

″db2iupgrade - Upgrade instance ″ in Command Reference

Some response file keywords have been deprecated

MIGRATE_PRIOR_VERSIONS and the **CONFIG_ONLY** response file keywords have been deprecated to reflect changes in Version 9.7 functionality and might be removed in a future release.

Details

The following response file keywords are no longer recommended:

- **MIGRATE_PRIOR_VERSIONS**
- **CONFIG_ONLY**

User response

Deprecated response file keyword	Change
MIGRATE_PRIOR_VERSIONS	Starting with DB2 V9.7, on Windows operating systems and non-root upgrades on Linux and UNIX operating systems, use the response file keyword UPGRADE_PRIOR_VERSIONS to specify an existing version of the DB2 product to be upgraded. Any existing response files should be changed to replace the deprecated keyword with the new keyword.
CONFIG_ONLY	Previously, on Linux and UNIX operating systems only, this keyword specified that the response file was for performing configuration tasks only. For example, creating a new instance using the db2isetup command. The keyword had two options: YES or NO. Starting with DB2 V9.7, there is no ability to set this option to NO. Whether the keyword is specified or not in the silent mode of db2isetup, the DB2 code assumes the option to be YES indicating that the response file is performing configuration tasks only. Existing scripts or response files do not need to be modified. Even if this keyword exists in a response file of db2isetup, the keyword value is assumed to be YES, regardless of the value specified.

Related concepts

“UPGRADE_PRIOR_VERSIONS response file keyword has been added” on page 121

“INTERACTIVE response file keyword has been changed” on page 144

Related reference

″Response file keywords″ in Installing DB2 Servers

Chapter 18. Discontinued functionality

Discontinued functionality is functionality that is no longer available. You need to make changes if you were using that functionality in previous releases.

The following DB2 functionality is no longer supported:

- Type-1 indexes (see "Type-1 indexes have been discontinued")
- Windows 32-bit partitioned databases (see "32-bit partitioned databases are no longer supported (Windows)" on page 188)
- Netscape browser support (see "Netscape browser support has been discontinued" on page 188)

The following products are no longer supported:

- XML Extender (see "XML Extender has been discontinued" on page 189)
- Web Object Runtime Framework (see "Web Object Runtime Framework (WORF) support has been discontinued" on page 189)
- DB2 embedded application server (see "DB2 embedded application server (EAS) has been discontinued" on page 190)

The following APIs, commands, command options, and registry variable are discontinued:

- db2uiddl command (see "db2uiddl command has been discontinued" on page 190)
- db2secv82 command (see "db2secv82 command has been discontinued" on page 190)
- GET AUTHORIZATIONS command (see "GET AUTHORIZATIONS command has been discontinued" on page 191)
- sqluadau API (see "sqluadau API and sql_authorization data structure have been discontinued" on page 191)
- **-a** and **-p** options of the db2ilist command (see "db2ilist command options -a and -p have been discontinued" on page 192)
- **DB2_THREAD_SUSPENSION** registry variable (see "Some registry and environment variables have been discontinued" on page 192)

Type-1 indexes have been discontinued

Type-1 indexes are no longer supported. You must convert type-1 indexes to type-2 indexes.

Details

All indexes that you created by using DB2 releases earlier than Version 8 are type-1 indexes, unless you converted them to type-2 indexes in Version 8 or later via the REORG INDEXES command with the **CONVERT** option. All indexes that you created by using Version 8.2, Version 9.1, or Version 9.5 are type-2 indexes, unless they were created in an instance with the **DB2_INDEX_TYPE2** registry variable set to NO, or unless you created an index on a table that already had a type-1 index. In Version 9.7, all indexes that you create are type-2 indexes.

If you do not convert your type-1 indexes before upgrading a database, these indexes are marked as invalid during the upgrade. If you set the **indexrec** configuration parameter to RESTART, the indexes are rebuilt as type-2 indexes when you restart the database. Otherwise, the rebuild occurs when you first access a table, and you might experience an unexpected degradation in response time. The table is inaccessible until the index rebuild is completed.

Also, the following related functionality is deprecated and might be removed in a future release:

- The **CONVERT** option of the REORG INDEXES command
- The **DB2LOADQUERY_TYPE1_INDEXES** parameter of the db2LoadQueryOutputStruct data structure and of the db2LoadQueryOutputStruct64 data structure of the db2LoadQuery API
- The DB2REORG_CONVERT parameter of the db2ReorgStruct data structure of the db2Reorg API

User response

Before upgrading to DB2 Version 9.7, convert type-1 indexes to type-2 indexes. Ensure that you allocate enough time to convert all the indexes prior to upgrading.

You can convert type-1 indexes to type-2 indexes by using the **CONVERT** option of the REORG INDEXES command or by using the output of the db2IdentifyType1 command. The db2IdentifyType1 command identifies and generates the appropriate statements that you can use later to convert any type-1 indexes found in tables or schemas for a specified database. For more information, see the “Converting type-1 indexes to type-2 indexes” topic.

Related reference

″db2IdentifyType1 - Type-1 index identification tool ″ in Command Reference

32-bit partitioned databases are no longer supported (Windows)

As of Version 9.7, partitioned databases are no longer supported on Windows 32-bit operating systems.

Details

Following the pervasive adoption and usage of 64-bit processors, the need for partitioned databases to run on 32-bit operating systems has been reduced.

User response

If you are using database partitioning functionality in Windows 32-bit environments, you will no longer be able to use these environments in Version 9.7. You can use database partitioning functionality in Windows 64-bit environments in Version 9.7.

Netscape browser support has been discontinued

Support of the Netscape browser is discontinued.

Details

In the past, you could use the Netscape browser to access DB2 Launchpad, First Steps, and the DB2 Information Center.

User response

Use one of the following supported browsers:

- Internet Explorer 6.0 and later
- Mozilla 1.7 and later
- Firefox 2.0 and later

The DB2 Launchpad supports also the SeaMonkey 1.1.4 (or later) browser.

The DB2 Information Center supports the same browsers, plus any other browsers that support JavaScript™.

Related concepts

"DB2 servers and IBM data server clients" in Installing DB2 Servers

XML Extender has been discontinued

As of Version 9.7, the functions provided by XML Extender have been replaced by the pureXML feature. As a result, XML Extender has been discontinued.

Details

Because the database provides a comprehensive set of XML tools, XML Extender functions are no longer required.

You can use the pureXML feature to store well-formed XML documents in database table columns that have the XML data type. By storing XML data in XML columns, the data is kept in its native hierarchical form, rather than stored as text or mapped to a different data model. You can apply database functions such as XMLQUERY and XSLTRANSFORM directly to database tables having the XML data type.

User response

Refer to the “Migrating from XML Extender to pureXML” topic for details on how to upgrade your existing database applications to Version 9.7 using the pureXML functionality.

Web Object Runtime Framework (WORF) support has been discontinued

The Web Objects Runtime Framework (WORF) has been discontinued. IBM Data Studio provides a simpler and more intuitive environment for quickly developing and deploying Web services.

Details

WORF has been replaced with a new feature within IBM Data Studio that you can use to create Web services without writing document access definition extension (DADX) files. Moreover, you can use the IBM Data Studio feature to create the SQL statements and stored procedures on which to base the operations of your Web services. Also, in many scenarios, deploying a Web service requires only a single mouse click.

User response

Migrate your WORF Web services to IBM Data Studio Web services. Instructions for migrating are in the “Migrating Web applications that were developed for the Web Object Runtime Framework (WORF)” topic in the Integrated Data Management Information Center located at http://publib.boulder.ibm.com/infocenter/idm/v2r2/index.jsp.

DB2 embedded application server (EAS) has been discontinued

DB2 EAS has been discontinued and the DB2WebServices application is not available in DB2 Version 9.7.

Details

DB2 EAS is no longer included with Version 9.7 products.

Resolution

Use IBM Data Studio or IBM Optim Development Studio to re-create your Web services and redeploy your Web services after upgrading to DB2 Version 9.7. If you have Web applications that were developed for WORF, you should migrate these Web applications. For more information, see “Web Object Runtime Framework (WORF) support has been discontinued” on page 189.

If you use DB2 samples that require an application server, you can use the WebSphere Application Server Community Edition (CE) application server.

db2uiddl command has been discontinued

The db2uiddl command has been discontinued because the conversion of unique indexes is handled during the conversion of type-1 indexes to type-2 indexes.

Details

The db2uiddl command generated scripts with CREATE UNIQUE INDEX statements to convert unique indexes created on databases earlier than DB2 UDB Version 5. This command is unnecessary because when you convert type-1 indexes to type-2 indexes, you also automatically convert the unique indexes created on databases earlier than DB2 UDB Version 5.

User response

Use the db2IdentifyType1 command to handle unique index conversion. For more information, see the *Converting type-1 indexes to type-2 indexes* topic.

Related reference

″db2IdentifyType1 - Type-1 index identification tool ″ in Command Reference

db2secv82 command has been discontinued

The db2secv82 command has been discontinued and has been replaced with the db2extsec command.

Details

Use the db2extsec command instead for setting the permissions for DB2 objects such as files, directories, network shares, registry keys, and services.

User response

Change references to the db2secv82 command in applications and scripts to the db2extsec command.

Related reference

″db2extsec - Set permissions for DB2 objects ″ in Command Reference

GET AUTHORIZATIONS command has been discontinued

The GET AUTHORIZATIONS command has been discontinued and replaced with the AUTH_LIST_AUTHORITIES_FOR_AUTHID table function.

Details

In previous releases, the GET AUTHORIZATIONS command reported the authorities of the current user from values found in the database configuration file and the authorization system catalog view (SYSCAT.DBAUTH). The command was deprecated in Version 9.5 because of internal changes in the DB2 authorization model and is discontinued in Version 9.7.

User response

Remove references to the GET AUTHORIZATIONS command, and use the AUTH_LIST_AUTHORITIES_FOR_AUTHID table function to return the authorities for a particular user.

sqluadau API and sql_authorization data structure have been discontinued

The sqluadau API and the sql_authorization data structure have been discontinued. You can use the AUTH_LIST_AUTHORITIES_FOR_AUTHID table function instead.

Details

In previous releases, the sqluadau API reported the instance level and database level authorities of the current user from values found in the database manager configuration file and the authorization system catalog view (SYSCAT.DBAUTH) respectively. This API has been deprecated since Version 9.5 due to internal changes in the DB2 authorization model and is discontinued in Version 9.7. The sql_authorization data structure has been discontinued because it was used only to return information after a call to the sqluadau API.

User response

Remove references to the sqluadau API and the sql_authorization data structure. Use the AUTH_LIST_AUTHORITIES_FOR_AUTHID table function to return the authorities for a particular user.

You can review the complete list of changed APIs in the “Changed APIs and data structures” topic.

Related reference

″Changed APIs and data structures″ in Administrative API Reference

db2ilist command options -a and -p have been discontinued

The **-a** and **-p** options of the db2ilist command have been discontinued.

Details

In DB2 Version 8, you could use the db2ilist command to list all of the DB2 instances that were available on a system. The **-a** and **-p** options were deprecated in DB2 Version 9.1 because the scope of the db2ilist command changed and listed only the instances related to the current installation path.

User response

For DB2 Version 9.1 and higher releases, you can list DB2 installation path information by using the db2ls command on the server and then the db2ilist command in each of the installation directories shown in the db2ls output.

For DB2 Version 8 releases, you can list instance information by running the db2ilist command in the installation directories where the DB2 Version 8 products are installed.

Related reference

″db2ls - List installed DB2 products and features ″ in Command Reference

Some registry and environment variables have been discontinued

The DB2_THREAD_SUSPENSION registry variable has been discontinued in Version 9.7.

The following registry variable is discontinued in Version 9.7:

Table 29. Registry variables discontinued in Version 9.7

Registry variable	Details
DB2_THREAD_SUSPENSION	This variable has been replaced with the DB2RESILIENCE variable which activates extended trap recovery by default. It also controls whether DB2 data page read errors are tolerated. For more information, see “Enhanced resilience to errors and traps reduces outages” on page 37

Related concepts

“Some registry and environment variables have been deprecated” on page 183

“Some registry and environment variables have changed” on page 136

Chapter 19. Version 9 deprecated and discontinued DB2 functionality summary

Due to changes in related functionality, the introduction of new functionality, or the removal of support, some DB2 for Linux, UNIX and Windows functionality available in earlier releases has been either deprecated or discontinued in Version 9.1, Version 9.5, or Version 9.7. Reviewing the summary of the changes will help you to understand the overall impact on your environment.

The functionality is grouped by the release in which the deprecation started. The information provided is cumulative: to get the complete list of deprecated functionality for a particular release, also review the information provided for the earlier releases:

- "Functionality deprecated in Version 8 and possibly discontinued in a later release" on page 194
- "Functionality deprecated in Version 9.1 and possibly discontinued in a later release" on page 196
- "Functionality deprecated in Version 9.5 and possibly discontinued in a later release" on page 199
- "Functionality deprecated in Version 9.7" on page 201

Note:

1. Pointers to supplemental information are provided if available.
2. Information about deprecated functionality for add-on features such as Spatial Extender is not included.
3. Information about deprecated registry variables related to functionality not described in other tables is listed separately.

To view the most up-to-date lists of discontinued functionality for the DB2 Version 9 releases, use the following information:

Table 30. V9 discontinued functionality

Release	Links to additional information
Version 9.1	• "Discontinued V9.1 functionality summary" at http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.rn.doc/doc/c0023234.htm • "V9.1 Deprecated and discontinued features" at http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/r0004670.htm
Version 9.5	• "Discontinued V9.5 functionality summary" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.wn.doc/doc/c0023234.html • "Some V9.5 registry and environment variables are discontinued" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.wn.doc/doc/i0052379.html
Version 9.7	• "Discontinued V9.7 functionality summary" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/c0023234.html

Functionality deprecated in Version 8 and possibly discontinued in a later release

Table 31. Functionality deprecated in Version 8 and possibly discontinued in a later release

Functionality	Discontinued in release	Links to additional information
Alternate FixPak images (UNIX operating systems)	Version 9.1	• Alternate FixPak images are discontinued (UNIX)
Audio, Image, and Video (AIV) Extenders	Version 9.1	• Audio, Image, and Video (AIV) Extenders are no longer supported
Autoloader utility (db2atld) and related registry variable	Version 9.1	• Autoloader utility (db2atld) is no longer supported
CALL_RESOLUTION DEFERRED statement and sqleproc API	To be determined	• Version 8 incompatibilities with previous releases
COLNAMES column in SYSCAT.INDEXES	To be determined	• DB2® Universal Database™ planned incompatibilities
db2profc command	Version 9.1	• db2sqljcustomize - DB2 SQLJ Profile Customizer Command • db2profc and db2profp utilities are discontinued
Data Links Manager, related registry variables and configuration parameters	Version 9.1	• Data Links Manager is no longer supported
Data Warehouse Center and Information Catalog Center	Version 9.1	• The Data Warehouse Center and the Information Catalog Center are no longer included
DB2 Administration Tools on some platforms	Version 9.1	• DB2 Administration Tools are no longer supported on some platforms
db2profp command	Version 9.1	• db2sqljprint - DB2 SQLJ Profile Printer Command • db2profc and db2profp utilities are discontinued
db2reg2large utility for converting DMS table space size	Version 9.1	• db2reg2large utility for converting DMS table space size is discontinued
Desktop icon and folder-making utilities (Linux operating systems)	Version 9.1	• Desktop icon and folder making utilities are no longer supported (Linux)
Extended Storage option for buffer pools	Version 9.1	• Extended Storage option for buffer pools is discontinued
JDBC type 2 driver	To be determined	• Introduction to Java application support
JDBC type 3 driver and related registry variable	Version 9.1	• Java Database Connectivity (JDBC) • Type 3 JDBC support is discontinued
NetBIOS and SNA communication protocols and related registry variables and configuration parameters	Version 9.1	• NetBIOS and SNA communication protocols are no longer supported

Table 31. Functionality deprecated in Version 8 and possibly discontinued in a later release (continued)

Functionality	Discontinued in release	Links to additional information
Network monitor elements: • **max_network_time_2_ms** • **max_network_time_8_ms** • **max_network_time_32_ms** • **max_network_time_gt32_ms**	To be determined	• Some network time monitor elements have been deprecated
PK_COLNAMES and FK_COLNAMES in SYSCAT.REFERENCES	To be determined	• DB2 Universal Database planned incompatibilities
Subset of ODBC 3.0 functions that affect CLI support	To be determined	• For the list of deprecated features and additional information, see CLI and ODBC function summary
Text Extender	Version 9.1	• Text Extender is no longer supported
Uncataloged stored procedures	To be determined	• Version 8 incompatibilities with previous releases
Variable argument list stored procedures	To be determined	• Version 8 incompatibilities with previous releases
Vendor escape clauses in CLI statements	To be determined	• Vendor escape clauses in CLI applications
Vendor load API (sqluvtld)	Version 9.1	• Vendor load API (sqluvtld) is discontinued
Virtual interface (VI) architecture usage in fast communications manager (FCM) and related registry variables	Version 9.1	• FCM no longer uses VI architecture
VSE and VM objects in the DB2 Control Center	Version 9.1	• VSE and VM objects are no longer supported in the DB2 Control Center

Table 32. Registry variables deprecated in Version 8 and possibly discontinued in a later release

Registry variable	Discontinued in release	Links to additional information
DB2_CLIENT_ENCALG	Version 9.1	• Deprecated and discontinued features
DB2JVIEW	Version 9.1	• Deprecated and discontinued features
DB2_LGPAGE_BP	Version 9.5	• List of Version 8 known limitations, problems, and workarounds • Some registry and environment variables are discontinued
DB2NOLIOAIO	Version 9.5	• DB2NOLIOAIO registry variable replaced by DB2LINUXAIO (Linux) • Some registry and environment variables are discontinued
DB2NTNOCACHE	To be determined	• Some registry and environment variables are discontinued
DB2_SCATTERED_IO (Linux)	Version 9.1	• DB2_SCATTERED_IO registry variable is discontinued (Linux)

Functionality deprecated in Version 9.1 and possibly discontinued in a later release

Table 33. Functionality deprecated in Version 9.1 and possibly discontinued in a later release

Functionality	**Discontinued in release**	**Links to additional information**
ADD PARTITIONING KEY and DROP PARTITIONING KEY clauses of the ALTER TABLE statement	To be determined	• ADD PARTITIONING KEY clause of the ALTER TABLE statement is deprecated • DROP PARTITIONING KEY clause of the ALTER TABLE statement is deprecated
Address Windowing Extensions (AWE) support and related registry variable (Windows operating systems)	Version 9.5	• Address Windowing Extensions (AWE) support is deprecated (Windows) • AWE feature support is discontinued (Windows)
bitwidth support for the db2icrt, db2iupdt, and db2ilist commands	Version 9.5	• -w option for db2icrt, db2ilist, and db2iupdt is discontinued (Linux and UNIX)
CLISchema CLI keyword	Version 9.5	• CLISchema CLI keyword is discontinued
COLNAMES column in SYSCAT.INDEXES	To be determined	• COLNAMES column in SYSCAT.INDEXES is deprecated
Database logging using raw devices	To be determined	• Database logging using raw devices is deprecated
db2ilist command options **-a** and **-p** (Linux and UNIX operating systems)	Version 9.7	• db2ilist command options are deprecated (Linux and UNIX) • db2ilist command options -a and -p have been discontinued
db2licm command option**-n**	Version 9.5	• -n option of the db2licm command is discontinued
db2undgp command	Version 9.5	• Deprecated and discontinued features • db2undgp command is discontinued
db2sampl command option**-schema**	Version 9.1, Fix Pack 2	• The -schema option of the db2sampl command is discontinued
db2secv82 command	Version 9.7	• db2secv82 command is deprecated • db2secv82 command has been discontinued
Default function entry points support in external routine libraries	To be determined	• External routines now require an explicit entry point specification
DB2 Web Tools	Version 9.5	• DB2 Web Tools support is deprecated • DB2 Web Tools support is discontinued
Extended storage (ESTORE) feature support, including related configuration parameters, monitor elements, and table functions	Version 9.5	• Extended storage (ESTORE) feature support is discontinued
iCheckPending parameter	To be determined	• Check pending table state is replaced and iCheckPending parameter is deprecated
Network Information Services (NIS and NIS+) support and related registry variable (Linux and UNIX operating systems)	To be determined	• Network Information Services (NIS and NIS+) support is deprecated (Linux and UNIX)

Table 33. Functionality deprecated in Version 9.1 and possibly discontinued in a later release (continued)

Functionality	Discontinued in release	Links to additional information
priv_mem_thresh configuration parameter	Version 9.5	• Deprecated and discontinued features • Some database manager configuration parameters have changed
Subset of SQL administrative routines	To be determined	• Deprecated Version 9.1 SQL administrative routines and their replacement routines or views • Deprecated Version 9.5 SQL administrative routines and their replacement routines or views • Deprecated Version 9.7 SQL administrative routines and their replacement routines or views
Type-1 indexes and related functionality	Version 9.7	• Type-1 indexes are deprecated • Type-1 indexes have been discontinued • db2uiddl command has been discontinued

Table 34. Registry variables deprecated in Version 9.1 and possibly discontinued in a later release

Registry variable	Discontinued in release	Links to additional information
DB2_ASYNC_APPLY	Version 9.5	• Deprecated and discontinued features • Some registry and environment variables are discontinued
DB2CCMSRV	Version 9.5	• Deprecated and discontinued features • Some registry and environment variables are discontinued
DB2_COMMIT_ON_EXIT	To be determined	• Deprecated and discontinued features
DB2_CORRELATED_PREDICATES	To be determined	• Deprecated and discontinued features
DB2_ENABLE_BUFPD	To be determined	• Deprecated and discontinued features
DB2LINUXAIO	Version 9.5	• DB2LINUXAIO registry variable is deprecated (Linux • Some registry and environment variables are discontinued
DB2_MAPPED_BASE	To be determined	• Deprecated and discontinued features
DB2_NO_MPFA_FOR_NEW_DB	To be determined	• Deprecated and discontinued features
DB2_PRED_FACTORIZE	To be determined	• Deprecated and discontinued features
OS/2® related registry variables: • **DB2UPMPR** • **DB2UPMSINGLE**	Version 9.5	• Deprecated and discontinued features • Some registry and environment variables are discontinued

Table 34. Registry variables deprecated in Version 9.1 and possibly discontinued in a later release (continued)

Registry variable	Discontinued in release	Links to additional information
Query Patroller registry variables: • **DQP_ABORTRESULT** • **DQP_CHILD_WAIT** • **DQP_DISKMON** • **DQP_EXIT_AN** • **DQP_INTERVAL** • **DQP_LAST_RESULT_DEST** • **DQP_LOCAL_SERVANTS** • **DQP_LOG** • **DQP_LOGMON** • **DQP_MAIL** • **DQP_MAIL_ACCOUNT** • **DQP_MAPI_PASSWORD** • **DQP_MAPI_PROFILE** • **DQP_NET** • **DQP_NOCPU** • **DQP_NOEXPLAIN** • **DQP_NTIER** • **DQP_PURGEHOURS** • **DQP_RECOVERY_INTERVAL** • **DQP_RES_TBLSPC** • **DQP_RUNTIME** • **DQP_SERVER** • **DQP_SHARE** • **DQP_SIBLING_WAIT** • **DQP_STARTUP** • **DQP_TRACEFILE**	To be determined	• Deprecated and discontinued features
DB2_BLOCK_ON_LOG_DISK_FULL	Version 9.5	• Deprecated and discontinued features • Some registry and environment variables are discontinued
DB2_FORCE_FCM_BP	Version 9.5	• Deprecated and discontinued features • Some registry and environment variables are discontinued
DB2_LGPAGE_BP	Version 9.5	• Deprecated and discontinued features • Some registry and environment variables are discontinued
DB2_MEMALLOCATE_HIGH	Version 9.5	• Deprecated and discontinued features • Some registry and environment variables are discontinued
DB2_MIGRATE_TS_INFO	Version 9.5	• Deprecated and discontinued features • Some registry and environment variables are discontinued

Table 34. Registry variables deprecated in Version 9.1 and possibly discontinued in a later release (continued)

Registry variable	Discontinued in release	Links to additional information
DB2_NEWLOGPATH2	Version 9.5	- Deprecated and discontinued features - Some registry and environment variables are discontinued
DB2_NR_CONFIG	Version 9.5	- Deprecated and discontinued features - Some registry and environment variables are discontinued
DB2_OLAP_BUFFER_SIZE	Version 9.5	- Deprecated and discontinued features - Some registry and environment variables are discontinued

Functionality deprecated in Version 9.5 and possibly discontinued in a later release

Table 35. Functionality deprecated in Version 9.5 and possibly discontinued in a later release

Functionality	Discontinued in release	Links to additional information
agentpri configuration parameter	To be determined	- Some database manager configuration parameters have changed
app_ctl_heap_sz, **appgroup_mem_sz**, and **groupheap_ratio** configuration parameters	Version 9.5	- Some database configuration parameters have changed
DB2 embedded application server (EAS)	Version 9.7	- DB2 embedded application server (EAS) has been discontinued
GET AUTHORIZATIONScommand	Version 9.7	- GET AUTHORIZATIONS command is deprecated - GET AUTHORIZATIONS command has been discontinued
IMPORT command options **CREATE** and **REPLACE_CREATE**	To be determined	- IMPORT command options CREATE and REPLACE_CREATE are deprecated
logretain and **userexit** configuration parameter	Version 9.5	- Some database configuration parameters have changed
Log control file SQLOGCTL.LFH	Version 9.5	- Log control file SQLOGCTL.LFH has been renamed and copied
LONG VARCHAR and LONG VARGRAPHIC data types	To be determined	- LONG VARCHAR and LONG VARGRAPHIC data types have been deprecated
maxagents and **maxcagents** configuration parameters	Version 9.5	- Some database manager configuration parameters have changed
Netscape browser support	Version 9.7	- Netscape browser support has been discontinued
numsegs configuration parameter	Version 9.5	- Some database configuration parameters have changed
query_heap_sz configuration parameter	Version 9.5	- Some database manager configuration parameters have changed

Table 35. Functionality deprecated in Version 9.5 and possibly discontinued in a later release (continued)

Functionality	Discontinued in release	Links to additional information
sqluadau API	Version 9.7	• sqluadau API is deprecated • sqluadau API and sql_authorization data structure have been discontinued
Static data stream snapshot output	To be determined	• Static data stream snapshot output is deprecated
Subset of SQL administrative routines	To be determined	• Deprecated Version 9.5 SQL administrative routines and their replacement routines or views • Deprecated Version 9.7 SQL administrative routines and their replacement routines or views
Web Object Runtime Framework (WORF) support	Version 9.7	• Web Object Runtime Framework (WORF) is deprecated • Web Object Runtime Framework (WORF) support has been discontinued
XML Extender	Version 9.7	• XML Extender is deprecated • XML Extender has been discontinued

Table 36. Registry variables deprecated in Version 9.5 and possibly discontinued in a later release

Registry variable	Discontinued in release	Links to additional information
DB2_ALLOCATION_SIZE	To be determined	• Some registry and environment variables are discontinued
DB2ATLD_PORTS	To be determined	• Some registry and environment variables are discontinued
DB2_ASYNC_IO_MAXFILOP	To be determined	• Some registry and environment variables are discontinued
DB2_BAR_AUTONOMIC_DISABLE	To be determined	• Some registry and environment variables are discontinued
DB2BPVARS	To be determined	• Some registry and environment variables are discontinued
DB2COUNTRY	To be determined	• Some registry and environment variables are discontinued
DB2DEFPREP	To be determined	• Some registry and environment variables are discontinued
DB2DMNBCKCTLR	To be determined	• Some registry and environment variables are discontinued
DB2FFDC	To be determined	• Some registry and environment variables are discontinued
DB2_HASH_JOIN	To be determined	• Some registry and environment variables are discontinued
DB2_INDEX_FREE	To be determined	• Some registry and environment variables are discontinued

Table 36. Registry variables deprecated in Version 9.5 and possibly discontinued in a later release (continued)

Registry variable	Discontinued in release	Links to additional information
DB2_MAP_XML_AS_CLOB_FOR_DLC	To be determined	• Some registry and environment variables are discontinued
DB2MEMMAXFREE	To be determined	• Some registry and environment variables are discontinued
DB2_NO_FORK_CHECK	To be determined	• Some registry and environment variables are discontinued
DB2_PARTITIONEDLOAD_DEFAULT	To be determined	• Some registry and environment variables are discontinued
DB2PRIORITIES and **DB2NTPRICLASS**	To be determined	• Some registry and environment variables are discontinued
DB2ROUTINE_DEBUG	To be determined	• Some registry and environment variables are discontinued
DB2_RR_TO_RS	To be determined	• Some registry and environment variables are discontinued
DB2_SNAPSHOT_NOAUTH	To be determined	• Some registry and environment variables are discontinued
DB2_TRUSTED_BINDIN	To be determined	• Some registry and environment variables are discontinued
DB2_UPDATE_PART_KEY	To be determined	• Some registry and environment variables are discontinued
DB2_VENDOR_INI	To be determined	• Some registry and environment variables are discontinued
DB2YIELD	To be determined	• Some registry and environment variables are discontinued

Functionality deprecated in Version 9.7

Table 37. Functionality deprecated in Version 9.7

Functionality	Links to additional information
CREATE EVENT MONITOR FOR DEADLOCKS statement and DB2DETAILDEADLOCK event monitor	• CREATE EVENT MONITOR FOR DEADLOCKS statement and DB2DETAILDEADLOCK event monitor have been deprecated
CREATE EVENT MONITOR FOR TRANSACTIONS event monitor	• CREATE EVENT MONITOR FOR TRANSACTIONS event monitor has been deprecated
DB2 Governor and Query Patroller	• DB2 Governor and Query Patroller have been deprecated
db2imigr, db2ckmig, and MIGRATE DATABASE commands; sqlemgdb and sqlgmgdb APIs	• Instance and database migration commands and APIs have been deprecated
db2iupdt command option -s	• db2iupdt command -s option has been deprecated (Linux and UNIX)
Health monitor	• Health monitor has been deprecated
LIST TABLESPACES and LIST TABLESPACE CONTAINERS commands	• LIST TABLESPACES and LIST TABLESPACE CONTAINERS commands have been deprecated

Table 37. Functionality deprecated in Version 9.7 (continued)

Functionality	Links to additional information
MIGRATE_PRIOR_VERSIONS and the **CONFIG_ONLY** response file keywords	• Some response file keywords have been deprecated
sqlugtpi API	• sqlugtpi API has been deprecated
sqlugrpn API	• sqlugrpn API has been deprecated
Subset of Net Search Extender features and commands	• Subset of Net Search Extender features and commands have been deprecated
Subset of SQL administrative routines	• Deprecated Version 9.7 SQL administrative routines and their replacement routines or views
The following Control Center tools: • Activity Monitor • Command Editor • Configuration Assistant • Control Center and associated wizards and advisors • Control Center plug-in extensions • DB2 administration server (DAS) • Event Analyzer • Health Center • Indoubt Transaction Monitor • Journal • License Center • Memory Visualizer • Query Patroller Center • Replication Center • Satellite Administration Center • Task Center	• Control Center tools and DB2 administration server (DAS) have been deprecated
Worksheet Format (WSF) for Export and Load utilities	• Worksheet Format (WSF) for Export and Load utilities has been deprecated

Table 38. Registry variables deprecated in Version 9.7

Registry variables	Links to additional information
DB2_CAPTURE_LOCKTIMEOUT	• Some registry and environment variables have been deprecated
DB2_SERVER_ENCALG	• Some registry and environment variables have been deprecated

Part 3. Appendixes

Appendix A. Overview of the DB2 technical information

DB2 technical information is available through the following tools and methods:

- DB2 Information Center
  - Topics (Task, concept and reference topics)
  - Help for DB2 tools
  - Sample programs
  - Tutorials
- DB2 books
  - PDF files (downloadable)
  - PDF files (from the DB2 PDF DVD)
  - printed books
- Command line help
  - Command help
  - Message help

Note: The DB2 Information Center topics are updated more frequently than either the PDF or the hardcopy books. To get the most current information, install the documentation updates as they become available, or refer to the DB2 Information Center at ibm.com.

You can access additional DB2 technical information such as technotes, white papers, and IBM Redbooks® publications online at ibm.com. Access the DB2 Information Management software library site at http://www.ibm.com/software/data/sw-library/.

Documentation feedback

We value your feedback on the DB2 documentation. If you have suggestions for how to improve the DB2 documentation, send an e-mail to db2docs@ca.ibm.com. The DB2 documentation team reads all of your feedback, but cannot respond to you directly. Provide specific examples wherever possible so that we can better understand your concerns. If you are providing feedback on a specific topic or help file, include the topic title and URL.

Do not use this e-mail address to contact DB2 Customer Support. If you have a DB2 technical issue that the documentation does not resolve, contact your local IBM service center for assistance.

DB2 technical library in hardcopy or PDF format

The following tables describe the DB2 library available from the IBM Publications Center at www.ibm.com/shop/publications/order. English and translated DB2 Version 9.7 manuals in PDF format can be downloaded from www.ibm.com/support/docview.wss?rs=71&uid=swg2700947.

Although the tables identify books available in print, the books might not be available in your country or region.

The form number increases each time a manual is updated. Ensure that you are reading the most recent version of the manuals, as listed below.

Note: The *DB2 Information Center* is updated more frequently than either the PDF or the hard-copy books.

Table 39. DB2 technical information

Name	Form Number	Available in print	Last updated
Administrative API Reference	SC27-2435-00	Yes	August, 2009
Administrative Routines and Views	SC27-2436-00	No	August, 2009
Call Level Interface Guide and Reference, Volume 1	SC27-2437-00	Yes	August, 2009
Call Level Interface Guide and Reference, Volume 2	SC27-2438-00	Yes	August, 2009
Command Reference	SC27-2439-00	Yes	August, 2009
Data Movement Utilities Guide and Reference	SC27-2440-00	Yes	August, 2009
Data Recovery and High Availability Guide and Reference	SC27-2441-00	Yes	August, 2009
Database Administration Concepts and Configuration Reference	SC27-2442-00	Yes	August, 2009
Database Monitoring Guide and Reference	SC27-2458-00	Yes	August, 2009
Database Security Guide	SC27-2443-00	Yes	August, 2009
DB2 Text Search Guide	SC27-2459-00	Yes	August, 2009
Developing ADO.NET and OLE DB Applications	SC27-2444-00	Yes	August, 2009
Developing Embedded SQL Applications	SC27-2445-00	Yes	August, 2009
Developing Java Applications	SC27-2446-00	Yes	August, 2009
Developing Perl, PHP, Python, and Ruby on Rails Applications	SC27-2447-00	No	August, 2009
Developing User-defined Routines (SQL and External)	SC27-2448-00	Yes	August, 2009
Getting Started with Database Application Development	GI11-9410-00	Yes	August, 2009
Getting Started with DB2 Installation and Administration on Linux and Windows	GI11-9411-00	Yes	August, 2009

Table 39. DB2 technical information (continued)

Name	Form Number	Available in print	Last updated
Globalization Guide	SC27-2449-00	Yes	August, 2009
Installing DB2 Servers	GC27-2455-00	Yes	August, 2009
Installing IBM Data Server Clients	GC27-2454-00	No	August, 2009
Message Reference Volume 1	SC27-2450-00	No	August, 2009
Message Reference Volume 2	SC27-2451-00	No	August, 2009
Net Search Extender Administration and User's Guide	SC27-2469-00	No	August, 2009
Partitioning and Clustering Guide	SC27-2453-00	Yes	August, 2009
pureXML Guide	SC27-2465-00	Yes	August, 2009
Query Patroller Administration and User's Guide	SC27-2467-00	No	August, 2009
Spatial Extender and Geodetic Data Management Feature User's Guide and Reference	SC27-2468-00	No	August, 2009
SQL Procedural Languages: Application Enablement and Support	SC27-2470-00	Yes	August, 2009
SQL Reference, Volume 1	SC27-2456-00	Yes	August, 2009
SQL Reference, Volume 2	SC27-2457-00	Yes	August, 2009
Troubleshooting and Tuning Database Performance	SC27-2461-00	Yes	August, 2009
Upgrading to DB2 Version 9.7	SC27-2452-00	Yes	August, 2009
Visual Explain Tutorial	SC27-2462-00	No	August, 2009
What's New for DB2 Version 9.7	SC27-2463-00	Yes	August, 2009
Workload Manager Guide and Reference	SC27-2464-00	Yes	August, 2009
XQuery Reference	SC27-2466-00	No	August, 2009

Table 40. DB2 Connect-specific technical information

Name	Form Number	Available in print	Last updated
Installing and Configuring DB2 Connect Personal Edition	SC27-2432-00	Yes	August, 2009
Installing and Configuring DB2 Connect Servers	SC27-2433-00	Yes	August, 2009

Table 40. DB2 Connect-specific technical information (continued)

Name	Form Number	Available in print	Last updated
DB2 Connect User's Guide	SC27-2434-00	Yes	August, 2009

Table 41. Information Integration technical information

Name	Form Number	Available in print	Last updated
Information Integration: Administration Guide for Federated Systems	SC19-1020-02	Yes	August, 2009
Information Integration: ASNCLP Program Reference for Replication and Event Publishing	SC19-1018-04	Yes	August, 2009
Information Integration: Configuration Guide for Federated Data Sources	SC19-1034-02	No	August, 2009
Information Integration: SQL Replication Guide and Reference	SC19-1030-02	Yes	August, 2009
Information Integration: Introduction to Replication and Event Publishing	GC19-1028-02	Yes	August, 2009

Ordering printed DB2 books

About this task

If you require printed DB2 books, you can buy them online in many but not all countries or regions. You can always order printed DB2 books from your local IBM representative. Keep in mind that some softcopy books on the *DB2 PDF Documentation* DVD are unavailable in print. For example, neither volume of the *DB2 Message Reference* is available as a printed book.

Printed versions of many of the DB2 books available on the DB2 PDF Documentation DVD can be ordered for a fee from IBM. Depending on where you are placing your order from, you may be able to order books online, from the IBM Publications Center. If online ordering is not available in your country or region, you can always order printed DB2 books from your local IBM representative. Note that not all books on the DB2 PDF Documentation DVD are available in print.

Note: The most up-to-date and complete DB2 documentation is maintained in the DB2 Information Center at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7.

To order printed DB2 books:

- To find out whether you can order printed DB2 books online in your country or region, check the IBM Publications Center at http://www.ibm.com/shop/publications/order. You must select a country, region, or language to access publication ordering information and then follow the ordering instructions for your location.

- To order printed DB2 books from your local IBM representative:
  1. Locate the contact information for your local representative from one of the following Web sites:
     - The IBM directory of world wide contacts at www.ibm.com/planetwide
     - The IBM Publications Web site at http://www.ibm.com/shop/publications/order. You will need to select your country, region, or language to the access appropriate publications home page for your location. From this page, follow the ″About this site″ link.
  2. When you call, specify that you want to order a DB2 publication.
  3. Provide your representative with the titles and form numbers of the books that you want to order. For titles and form numbers, see “DB2 technical library in hardcopy or PDF format” on page 205.

Displaying SQL state help from the command line processor

DB2 products return an SQLSTATE value for conditions that can be the result of an SQL statement. SQLSTATE help explains the meanings of SQL states and SQL state class codes.

To start SQL state help, open the command line processor and enter:

```
? sqlstate or ? class code
```

where *sqlstate* represents a valid five-digit SQL state and *class code* represents the first two digits of the SQL state.
For example, `? 08003` displays help for the 08003 SQL state, and `? 08` displays help for the 08 class code.

Accessing different versions of the DB2 Information Center

About this task

For DB2 Version 9.7 topics, the DB2 Information Center URL is http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/

For DB2 Version 9.5 topics, the DB2 Information Center URL is http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/

For DB2 Version 9 topics, the DB2 Information Center URL is http://publib.boulder.ibm.com/infocenter/db2luw/v9/

For DB2 Version 8 topics, go to the Version 8 Information Center URL at: http://publib.boulder.ibm.com/infocenter/db2luw/v8/

Displaying topics in your preferred language in the DB2 Information Center

About this task

The DB2 Information Center attempts to display topics in the language specified in your browser preferences. If a topic has not been translated into your preferred language, the DB2 Information Center displays the topic in English.

- To display topics in your preferred language in the Internet Explorer browser:

1. In Internet Explorer, click the **Tools** —> **Internet Options** —> **Languages...** button. The Language Preferences window opens.
2. Ensure your preferred language is specified as the first entry in the list of languages.
   - To add a new language to the list, click the **Add...** button.

     Note: Adding a language does not guarantee that the computer has the fonts required to display the topics in the preferred language.
   - To move a language to the top of the list, select the language and click the **Move Up** button until the language is first in the list of languages.
3. Clear the browser cache and then refresh the page to display the DB2 Information Center in your preferred language.

- To display topics in your preferred language in a Firefox or Mozilla browser:
  1. Select the button in the **Languages** section of the **Tools** —> **Options** —> **Advanced** dialog. The Languages panel is displayed in the Preferences window.
  2. Ensure your preferred language is specified as the first entry in the list of languages.
     - To add a new language to the list, click the **Add...** button to select a language from the Add Languages window.
     - To move a language to the top of the list, select the language and click the **Move Up** button until the language is first in the list of languages.
  3. Clear the browser cache and then refresh the page to display the DB2 Information Center in your preferred language.

Results

On some browser and operating system combinations, you must also change the regional settings of your operating system to the locale and language of your choice.

Updating the DB2 Information Center installed on your computer or intranet server

A locally installed DB2 Information Center must be updated periodically.

Before you begin

Before you begin

A DB2 Version 9.7 Information Center must already be installed. For details, see the "Installing the DB2 Information Center using the DB2 Setup wizard" topic in *Installing DB2 Servers*. All prerequisites and restrictions that applied to installing the Information Center also apply to updating the Information Center.

About this task

About this task

An existing DB2 Information Center can be updated automatically or manually:

- Automatic updates - updates existing Information Center features and languages. An additional benefit of automatic updates is that the Information

Center is unavailable for a minimal period of time during the update. In addition, automatic updates can be set to run as part of other batch jobs that run periodically.

- Manual updates - should be used when you want to add features or languages during the update process. For example, a local Information Center was originally installed with both English and French languages, and now you want to also install the German language; a manual update will install German, as well as, update the existing Information Center features and languages. However, a manual update requires you to manually stop, update, and restart the Information Center. The Information Center is unavailable during the entire update process.

Procedure

This topic details the process for automatic updates. For manual update instructions, see the "Manually updating the DB2 Information Center installed on your computer or intranet server" topic.

To automatically update the DB2 Information Center installed on your computer or intranet server:

1. On Linux operating systems,
   a. Navigate to the path where the Information Center is installed. By default, the DB2 Information Center is installed in the /opt/ibm/db2ic/V9.7 directory.
   b. Navigate from the installation directory to the doc/bin directory.
   c. Run the ic-update script:
      ```
      ic-update
      ```
2. On Windows operating systems,
   a. Open a command window.
   b. Navigate to the path where the Information Center is installed. By default, the DB2 Information Center is installed in the <Program Files>\IBM\DB2 Information Center\Version 9.7 directory, where `<Program Files>` represents the location of the Program Files directory.
   c. Navigate from the installation directory to the doc\bin directory.
   d. Run the ic-update.bat file:
      ```
      ic-update.bat
      ```

Results

Results

The DB2 Information Center restarts automatically. If updates were available, the Information Center displays the new and updated topics. If Information Center updates were not available, a message is added to the log. The log file is located in doc\eclipse\configuration directory. The log file name is a randomly generated number. For example, 1239053440785.log.

Manually updating the DB2 Information Center installed on your computer or intranet server

If you have installed the DB2 Information Center locally, you can obtain and install documentation updates from IBM.

About this task

Updating your locally-installed DB2 Information Center manually requires that you:

1. Stop the DB2 Information Center on your computer, and restart the Information Center in stand-alone mode. Running the Information Center in stand-alone mode prevents other users on your network from accessing the Information Center, and allows you to apply updates. The Workstation version of the DB2 Information Center always runs in stand-alone mode. .
2. Use the Update feature to see what updates are available. If there are updates that you must install, you can use the Update feature to obtain and install them

   Note: If your environment requires installing the DB2 Information Center updates on a machine that is not connected to the internet, mirror the update site to a local file system using a machine that is connected to the internet and has the DB2 Information Center installed. If many users on your network will be installing the documentation updates, you can reduce the time required for individuals to perform the updates by also mirroring the update site locally and creating a proxy for the update site.
   If update packages are available, use the Update feature to get the packages. However, the Update feature is only available in stand-alone mode.
3. Stop the stand-alone Information Center, and restart the DB2 Information Center on your computer.

Note: On Windows 2008, Windows Vista (and higher), the commands listed later in this section must be run as an administrator. To open a command prompt or graphical tool with full administrator privileges, right-click the shortcut and then select **Run as administrator**.

To update the DB2 Information Center installed on your computer or intranet server:

1. Stop the DB2 Information Center.
   - On Windows, click **Start** → **Control Panel** → **Administrative Tools** → **Services**. Then right-click **DB2 Information Center** service and select **Stop**.
   - On Linux, enter the following command:
     ```
     /etc/init.d/db2icdv97 stop
     ```
2. Start the Information Center in stand-alone mode.
   - On Windows:
     a. Open a command window.
     b. Navigate to the path where the Information Center is installed. By default, the DB2 Information Center is installed in the <Program Files>\IBM\DB2 Information Center\Version 9.7 directory, where `<Program Files>` represents the location of the Program Files directory.
     c. Navigate from the installation directory to the doc\bin directory.
     d. Run the help_start.bat file:
        ```
        help_start.bat
        ```
   - On Linux:
     a. Navigate to the path where the Information Center is installed. By default, the DB2 Information Center is installed in the /opt/ibm/db2ic/V9.7 directory.
     b. Navigate from the installation directory to the doc/bin directory.
     c. Run the help_start script:

```
    help_start
```

The systems default Web browser opens to display the stand-alone Information Center.

3. Click the **Update** button ( ). (JavaScript must be enabled in your browser.) On the right panel of the Information Center, click **Find Updates**. A list of updates for existing documentation displays.
4. To initiate the installation process, check the selections you want to install, then click **Install Updates**.
5. After the installation process has completed, click **Finish**.
6. Stop the stand-alone Information Center:
   - On Windows, navigate to the installation directory's doc\bin directory, and run the help_end.bat file:

     ```
     help_end.bat
     ```

     Note: The help_end batch file contains the commands required to safely stop the processes that were started with the help_start batch file. Do not use `Ctrl-C` or any other method to stop help_start.bat.
   - On Linux, navigate to the installation directory's doc/bin directory, and run the help_end script:

     ```
     help_end
     ```

     Note: The help_end script contains the commands required to safely stop the processes that were started with the help_start script. Do not use any other method to stop the help_start script.
7. Restart the DB2 Information Center.
   - On Windows, click **Start** → **Control Panel** → **Administrative Tools** → **Services**. Then right-click **DB2 Information Center** service and select **Start**.
   - On Linux, enter the following command:

     ```
     /etc/init.d/db2icdv97 start
     ```

Results

The updated DB2 Information Center displays the new and updated topics.

DB2 tutorials

The DB2 tutorials help you learn about various aspects of DB2 products. Lessons provide step-by-step instructions.

Before you begin

You can view the XHTML version of the tutorial from the Information Center at http://publib.boulder.ibm.com/infocenter/db2help/.

Some lessons use sample data or code. See the tutorial for a description of any prerequisites for its specific tasks.

DB2 tutorials

To view the tutorial, click the title.

“pureXML” in *pureXML Guide*

Set up a DB2 database to store XML data and to perform basic operations with the native XML data store.

“Visual Explain” in *Visual Explain Tutorial*

Analyze, optimize, and tune SQL statements for better performance using Visual Explain.

DB2 troubleshooting information

A wide variety of troubleshooting and problem determination information is available to assist you in using DB2 database products.

DB2 documentation

Troubleshooting information can be found in the *DB2 Troubleshooting Guide* or the Database fundamentals section of the *DB2 Information Center*. There you will find information about how to isolate and identify problems using DB2 diagnostic tools and utilities, solutions to some of the most common problems, and other advice on how to solve problems you might encounter with your DB2 database products.

DB2 Technical Support Web site

Refer to the DB2 Technical Support Web site if you are experiencing problems and want help finding possible causes and solutions. The Technical Support site has links to the latest DB2 publications, TechNotes, Authorized Program Analysis Reports (APARs or bug fixes), fix packs, and other resources. You can search through this knowledge base to find possible solutions to your problems.

Access the DB2 Technical Support Web site at http://www.ibm.com/software/data/db2/support/db2_9/

Terms and Conditions

Permissions for the use of these publications is granted subject to the following terms and conditions.

Personal use: You may reproduce these Publications for your personal, non commercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these Publications, or any portion thereof, without the express consent of IBM.

Commercial use: You may reproduce, distribute and display these Publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these Publications, or reproduce, distribute or display these Publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the Publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the Publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED ″AS-IS″ AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Appendix B. Notices

This information was developed for products and services offered in the U.S.A. Information about non-IBM products is based on information available at the time of first publication of this document and is subject to change.

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 grant 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 character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
3-2-12, Roppongi, Minato-ku, Tokyo 106-8711 Japan

The following paragraph does not apply to the United Kingdom or any other country/region where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

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

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

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

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 that has been exchanged, should contact:

IBM Canada Limited
Office of the Lab Director
8200 Warden Avenue
Markham, Ontario
L6G 1C7
CANADA

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.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements, or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

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

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious, and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

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

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (*your company name*) (*year*). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _*enter the year or years*_. All rights reserved.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

The following terms are trademarks or registered trademarks of other companies

- Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
- Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- Intel, Intel logo, Intel Inside®, Intel Inside logo, Intel® Centrino®, Intel Centrino logo, Celeron®, Intel® Xeon®, Intel SpeedStep®, Itanium®, and Pentium® are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
- Microsoft, Windows, Windows NT®, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Index

Printed in USA

SC27-2463-00

Spine information:

IBM DB2 9.7 for Linux, UNIX, and Windows

What’s New for DB2 Version 9.7