

DB2
for Linux, UNIX, and Windows



Version 9 Release 7



What's New for DB2 Version 9.7
Updated July, 2012

DB2
for Linux, UNIX, and Windows



Version 9 Release 7



What's New for DB2 Version 9.7
Updated July, 2012

Note

Before using this information and the product it supports, read the general information under Appendix C, "Notices," on page 357.

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.

© Copyright IBM Corporation 2012.

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

Contents

About this book	xi
Who should use this book	xi
How this book is structured	xi
Highlighting conventions	xiii
<hr/>	
Part 1. New features and functionality	1
<hr/>	
Chapter 1. Product packaging enhancements	3
Component names have changed	3
FP3: Advanced product capabilities.	3
Chapter 2. Compression enhancements	5
XML data stored in the XML storage object of tables can be compressed	5
Temporary tables can be compressed	6
Indexes can be compressed	7
Data replication source tables can be compressed	7
Chapter 3. Manageability enhancements	9
Existing databases can now use automatic storage	10
Automatic storage supports dropping storage paths and rebalancing table spaces	10
Reclaiming unused space from table spaces is supported	11
Reclaiming space from multidimensional clustering (MDC) tables has been enhanced	13
Enhanced access to DB2 administration commands through SQL	13
Table data can be moved online using a new stored procedure	14
Table space capacity for large and temporary table spaces has increased	15
Distribution map supports larger data warehouses	15
Database partition servers are online immediately after being added to an instance	16
DESCRIBE command provides additional index information	17
FP1: Data partitions and partitioned indexes can be reorganized.	17
FP1: relocating databases using the db2relocatedb command has been improved	18
FP2: New RESTRICTED ACCESS option restricts database connections within quiesced instance	18
FP4: db2look command enhances DDL statement generation for database objects and dependents	19
FP5: Access control, error checking, and table exclusion during data redistribution have been improved	19
Chapter 4. pureXML enhancements	21
Partitioned tables support XML data	22
MDC tables support XML columns	23
Declared temporary tables support XML columns	23
Inlined SQL functions support the XML data type	24
Partitioned database environments support the pureXML feature	25
FP6: XML data type support added in global variables and compiled SQL functions	26
Partitioned indexes on partitioned tables improve performance	27
DESCRIBE command provides additional index information	29
The ALTER TABLE statement supports REORG-recommended operations containing XML data	29
Predicate pushdown is available for SQL/XML and XQuery statements	30
XML parsing and validation can return more detailed messages.	30
Small LOBs can be stored in table rows and compressed	30
FP1: XQuery functions make it easier to retrieve date and time values for local time zones.	31
Multiple XML documents in a column can be decomposed at the same time	32
Optimization profiles support guidelines for XML data.	32
Write access is supported during creation or reorganization of indexes over XML data	33
FP1: Distribution statistics collected for XML columns	34
XML data stored in the XML storage object of tables can be compressed	35

Chapter 5. Monitoring enhancements	37
New relational monitoring interfaces are light weight and SQL accessible	38
Maximum number of active event monitors is increased	39
FP1: New relational monitoring interfaces for locking events	40
New monitor elements and database configuration parameters provide more granular monitoring	41
New relational interface to monitor dynamic and static SQL statements in package cache	42
A new unit of work event monitor supports transaction monitoring	43
Time-spent monitor elements are more comprehensive	44
FP1: Last referenced date is available for tables, table partitions, indexes, and packages	45
Lock event reporting has been enhanced	46
The section for SQL statements captured by activity event monitor can now be collected	46
FP1: Statements from a runtime section can be explained	47
FP1: Explain enhanced with actual values for operator cardinality	48
FP1: New event monitor for dynamic and static SQL statements in package cache.	48
Additional system monitoring information can be generated	49
FP1: Text reports can be generated based on monitoring data.	50
FP1: Monitoring table functions information can be viewed using administrative views	50
FP1: Table functions for row-based formatting of monitoring information are available	51
FP2: Identify FCM issues more easily.	52
FP3: Simplify capture of detailed statement information using the new db2caem tool with db2support integration options	52
FP4: Simplified access to activity metrics in the activity event monitor.	53
FP5: New monitoring table functions return information about memory	54
FP6: New table functions provide SQL-based access to system information	54
FP6: New logical data groups added to the statistics event monitor.	55
FP6: XML document metrics stores system metrics collected by statistics event monitor.	56
Chapter 6. High availability, backup, logging, resiliency, and recovery enhancements	57
Cluster management software integration support has been extended (Solaris)	58
FP1: DB2 Advanced Copy Services (ACS) is supported on AIX 6.1	58
Enhanced resilience to errors and traps reduces outages	58
FP1: Read operations on HADR standby databases are supported	59
FP3: Data deduplication device support has been integrated into the backup utilities.	60
FP2: Proxy node support for the db2adutl command has been added	60
Administration notification and diagnostic logs occupy specified amount of disk space	60
FP2: Databases can be restored using transportable sets.	61
FP4: New scripts improve integration between DB2 High Availability Disaster Recovery and IBM Tivoli System Automation for Multiplatforms (Windows)	62
FP4: Roving high availability (HA) failover reduces downtime in partitioned database environments	63
FP5: Automatic recovery utilities now recognize merged backups	63
FP5: db2adutl enhancements simplify backup image and log file management	63
FP5: ADMIN_MOVE_TABLE stored procedure now supports recoverable load	64
FP5: Super asynchronous HADR synchronization mode has been added	64
Chapter 7. Performance enhancements.	67
Access plan reuse ensures consistent performance	68
Statement concentrator enables access plan sharing	68
Statistics collection sampling performance for statistical views has been improved.	69
Optimization guidelines for packages can be applied more easily	69
Cost model has been improved for queries in partitioned database environments	70
Cursor stability (CS) isolation level enhancements provide more concurrency	70
Scan sharing improves concurrency and performance	71
Partitioned indexes on partitioned tables improve performance	72
Materialized query table (MQT) matching includes more scenarios	74
Small LOBs can be stored in table rows and compressed	75
I/O completion ports (IOCP) is used by default for asynchronous I/O (AIX)	76
Isolation level locking intent can be specified in subselect and fullselect clauses	76
FP1: Partitioned table data remains available during roll-out operations	77
FP2: Materialized query tables with spatial columns can be replicated.	78

FP2: Additional optimization of performance using IBM InfoSphere Optim Performance Manager Extended Edition	78
FCM parallelism support added	78
Chapter 8. SQL compatibility enhancements	81
SQL*Plus compatible command line processor has been added	81
Oracle data dictionary-compatible view support has been added	82
PL/SQL language is supported by DB2 interfaces	82
Data type support has been extended.	83
Alternative SQL language syntax is supported.	84
Oracle-compatible mode can be enabled.	84
FP4: CREATE TRIGGER statement enhancements.	85
FP5: SQL compatibility has been enhanced	85
Chapter 9. Workload management enhancements	87
Workloads support activity-based threshold controls.	88
Workload connection attributes have been enhanced.	88
Workload-level aggregate activity data collection and new high watermarks provide additional statistics	89
Buffer pool I/O priority can be controlled for service classes	90
DB2 workload manager supports integration with Linux workload management (WLM)	90
New thresholds provide additional activity control	91
Time-based thresholds support finer granularity	91
Support of priority aging of in-progress activities has been added	92
FP1: Work action sets can be defined at workload level.	94
FP1: New time threshold limits unit of work duration	94
FP1: Script facilitates migration from Query Patroller to workload manager	95
Chapter 10. Security enhancements	97
DB2 authorization model has been enhanced to allow separation of duties	98
AES encryption of user ID and password enhances security	100
SSL client support expanded and configuration simplified	101
FP1: Transparent LDAP authentication and group lookup is supported (Linux and UNIX).	103
Passwords can be the maximum length supported by the operating system	103
SYSMON authority has been extended to LIST commands and the db2mtrk command	104
Fenced mode process (db2fmp) privileges can be customized (Windows)	104
FP1: 32-bit GSKit libraries are included in the 64-bit DB2 product installation	105
FP2: Audit improvements allow replay of past database activities	105
Chapter 11. Application development enhancements	107
In-database analytics with SAS embedded process support added	109
Columns can be renamed using the ALTER TABLE statement	110
REPLACE added as an option on several CREATE statements	110
Transactions can include an increased limit of ALTER TABLE operations	111
CREATE with errors support has been added for certain database objects	111
Soft invalidation and automatic revalidation support added	112
ALTER COLUMN SET DATA TYPE support has been extended	114
The SELECT INTO statement supports the FOR UPDATE clause	115
TRUNCATE added as an SQL statement	115
Created temporary tables are supported	116
Scalar function support has been extended	116
Implicit casting simplifies application enablement	119
TIMESTAMP data type allows for parameterized precision	120
Temporary tables support LOB-type columns.	120
Public aliases simplify global object referencing	121
Default values and named arguments promote flexibility for creating and calling procedures and functions	121
Autonomous transactions are supported	122
FP3: Application environments can be customized during the connection process	123
Python application development support has been added	123
System-defined modules simplify SQL PL and application logic	124
Common SQL API supports the development of portable administrative applications	125

IBM Database Add-Ins for Visual Studio have been enhanced	126
New DB2 sample programs have been added	127
FP1: User-defined functions support OUT and INOUT parameters	130
FP3: Full support for db2dsdriver configuration file extended to IBM data server clients for CLI and open source applications	130
FP4: New IBM Data Server Driver keyword for a finer grained timeout value	131
FP6: Environment variables now contain the driver path (Windows)	131
FP6: New utility validates a Net Search Extender text index (Windows, AIX)	131
IBM data server clients and drivers enhancements	131
JDBC and SQLJ support has been enhanced	132
IBM Data Server Driver Package has been enhanced	148
Trusted context support has been extended	149
Sysplex support is extended to IBM data server clients and non-Java data server drivers	149
Call level interface (CLI) functionality has been enhanced	150
IBM Data Server Provider for .NET is enhanced	160
Chapter 12. SQL Procedural Language (SQL PL) enhancements	169
Related database object definitions can be encapsulated in new module database object	169
Compiled compound statement support has been added	170
Triggers support has been extended	171
SQL PL functionality has been extended for user-defined functions	171
FP1: Global variable assignments in nested contexts are supported	173
New data types are supported.	173
Anchored data type has been added.	173
Boolean data type has been added	174
Associative array data type has been added	175
Cursor data type support has been added	175
Row data type has been added	176
Chapter 13. DB2 Text Search and Net Search Extender enhancements.	179
Full-text searches support partitioned tables	179
Full-text searches support in partitioned database environments has been extended	179
Incremental update based on integrity processing is supported	180
FP3: DB2 Text Search and Net Search Extender index coexistence	180
Chapter 14. Installation, upgrade, and fix pack enhancements	183
Instances and DB2 Administration Server can be created in a shared DB2 copy (Linux and UNIX)	184
db2rspgn command supported on Linux and UNIX operating systems	184
Uninstalling using a response file is supported in more situations	185
New response file keywords have been added	185
Product installation can be validated using db2val command	186
Product update service support has been extended	186
Product installation on Linux and UNIX platforms has been enhanced	187
FP3: 32bit IBM Database Add-ins for Visual Studio is now available with the 64-bit DB2 installation images.	188
IBM Tivoli System Automation for Multiplatforms (SA MP) support has been improved	188
Installation images can be reduced (Linux and UNIX)	188
Universal fix packs support has been extended (Windows)	189
Fix pack installations can require less space (Linux and UNIX)	189
FP3: DB2 Connect Unlimited Edition for System z license activation process has been simplified	189
FP4: Installation fix pack support has been extended	190
FP5: IBM Database Enterprise Developer Edition product bundle extended.	190
Chapter 15. Multicultural support enhancements	191
GB18030 code set support has been extended.	191
Chapter 16. Troubleshooting and problem determination enhancements	193
FP5: Diagnosing upgrade problems is easier	193
FP5: Table space modification status can be checked	193
FP5: db2trc command has been improved for clients	194

FP5: First occurrence data collection supports new collection types and collection triggered by user-defined thresholds	194
FP5: Load serviceability has been improved	195
FP5: db2dart command has extended functionality that helps improve performance.	196
FP4: The serviceability of large database systems has improved	196
FP4: New configuration parameter reduces risk of losing diagnostic data	200
FP4: Archive log files can be checked for validity	201
FP4: DB2 Text Search infrastructure improvements	201
FP3: Improvements to the granularity of the db2trc tool	201
FP1: Diagnostic data can be stored in separate directories	202
FP1: db2support tool has been enhanced	203
FP1: Fenced routines history information is easier to collect	203
Administration notification and diagnostic logs occupy specified amount of disk space.	204

Part 2. DB2 Connect enhancements and changes summary 207

Chapter 17. DB2 Connect Version 9.7 fix pack summary 211

Chapter 18. FP4: Installation fix pack support has been extended. 217

Part 3. What's changed 219

Chapter 19. Changed functionality 221

Administration changes summary	221
Partitioned indexes are created by default for partitioned tables	221
Some database manager configuration parameters have been changed	222
FP1: Collection interval time for workload management statistics has changed	224
NO FILE SYSTEM CACHING for table space containers is the default for General Parallel File System (GPFS)	224
Some registry and environment variables have changed	225
Primary and secondary log files use non-buffered I/O by default	231
AUTOCONFIGURE command has been changed	231
The CONCURRENTDBCOORDACTIVITIES threshold has been changed	232
DESCRIBE command lists information about additional index types	233
FP1: Detach operation for data partitions has been changed	233
FP1: XML schema maxOccurs attribute values greater than 5000 are parsed differently	234
FP5: History file no longer locked during automatic deletion of recovery objects	235
FP5: Some administrative routines and views have changed	235
Database setup and product installation changes summary	236
Licensing control for DB2 Express, DB2 Workgroup Edition, and workload management has been changed	237
License enforcement policies list has been updated	237
Some database configuration parameters have been changed	238
INTERACTIVE response file keyword has been changed	241
Registry files have been removed from the DB2 installation path	241
DB2 Text Search installation has changed	242
FP2: DB2 Advanced Copy Services (ACS) is not automatically included in a compact installation	242
Security changes summary	243
System administrator (SYSADM) authority scope has changed	243
Security administrator (SECADM) abilities have been extended	245
Database administrator (DBADM) authority scope has changed	246
SSLconfig.ini and SSLClientconfig.ini files replaced with new database manager configuration parameters	247
Audit stored procedures and table functions now require only the EXECUTE privilege.	248
Net Search Extender command authorizations have changed	249
DB2 Text Search command and stored procedure authorizations have changed	250
Application development changes summary	252
Default value or behavior of selected IBM Data Server Driver keywords have changed.	252
Default values of selected automatic client reroute IBM Data Server Driver keywords have changed	252
Cursor stability (CS) level behavior for newly created databases has been changed	253
Maximum limit of log sequence numbers has increased	254
Some system catalog views, system-defined administrative routines and views have been added and changed	254

New SYSIBM functions override unqualified user-defined functions with the same name	263
Untyped NULL keyword specifications no longer resolve to identifier names	264
CHAR(<i>decimal-expression</i>) scalar function return behavior has been changed	265
DOUBLE(<i>string-expression</i>) scalar function return behavior has been changed	266
Result data type for unary minus and unary operators in untyped expressions has been changed	267
DEFAULT keyword specification has been changed	267
XML data is passed by reference in SQL stored procedures	268
Type annotations for validated XML documents are unavailable	270
Merge modules for ODBC, CLI, and .NET have been combined (Windows)	271
Result data type for integer division in number_compat mode has been changed.	272
FP1: Some import and load file type modifiers can accept unspecified values in trailing fields	272
Routines with bulk collection operations need to be re-created	273

Chapter 20. Deprecated functionality in Version 9.7 275

Control Center tools have been deprecated	276
DB2 administration server (DAS) has been deprecated.	286
DB2 Governor and Query Patroller have been deprecated	286
Health monitor has been deprecated.	288
FP4: DB2 Health Advisor has been deprecated	289
FP5: IBM DB2 Geodetic Data Management Feature software has been deprecated	290
FP5: Microsoft Visual Studio 2005 support has been deprecated	290
LONG VARCHAR and LONG VARGRAPHIC data types have been deprecated	291
Worksheet Format (WSF) for Import and Export utilities has been deprecated	291
LIST TABLESPACES and LIST TABLESPACE CONTAINERS commands have been deprecated	292
IBM Software Developer's Kit (SDK) 1.4.2 support for Java routines has been deprecated	292
sqlugrpn API has been deprecated	293
sqlugtpi API has been deprecated	293
DB2SE_USA_GEOCODER has been deprecated	294
Subset of Net Search Extender features and commands have been deprecated	294
Some registry and environment variables have been deprecated	295
FP1: -file option of db2rfrpn command has been deprecated.	296
FP2: COBOL and FORTRAN language support has been deprecated for db2History APIs	297
FP3: HP-UX 32-bit client support has been deprecated.	297
CREATE EVENT MONITOR FOR DEADLOCKS statement and DB2DETAILDEADLOCK event monitor have been deprecated	298
CREATE EVENT MONITOR FOR TRANSACTIONS statement has been deprecated.	298
db2iupdt command -s option has been deprecated (Linux and UNIX)	299
Instance and database migration commands and APIs have been deprecated	299
Some response file keywords have been deprecated	300
FP4: Distributed installation support with Microsoft Systems Management Server is deprecated (Windows)	301
FP4: The -global option for troubleshooting tools has been deprecated	302
FP5: Some monitoring routines and views are deprecated	302
FP6: Reporting of metrics in details_xml by the statistics event monitor has been deprecated.	303

Chapter 21. Discontinued functionality in Version 9.7 305

Some operating systems are no longer supported	305
Type-1 indexes have been discontinued.	306
32-bit partitioned databases are no longer supported (Windows)	307
Netscape browser support has been discontinued	307
XML Extender has been discontinued	308
Web Object Runtime Framework (WORF) support has been discontinued	308
DB2 embedded application server (EAS) has been discontinued	309
IBM DB2 Everyplace products have been discontinued	309
db2uiddl command has been discontinued	309
db2secv82 command has been discontinued	310
GET AUTHORIZATIONS command has been discontinued	310
sqladau API and sql_authorization data structure have been discontinued.	311
db2iilist command options -a and -p have been discontinued	311
Some registry and environment variables have been discontinued.	312

Chapter 22. Summary of deprecated and discontinued DB2 functionality in Version 9.7 and earlier releases	313
<hr/>	
Part 4. Appendixes	325
Appendix A. DB2 Version 9.7 for Linux, UNIX, and Windows fix pack summary	327
Appendix B. Overview of the DB2 technical information	345
DB2 technical library in hardcopy or PDF format	345
Ordering printed DB2 books	348
Displaying SQL state help from the command line processor	349
Accessing different versions of the DB2 Information Center	349
Displaying topics in your preferred language in the DB2 Information Center	350
Updating the DB2 Information Center installed on your computer or intranet server	351
Manually updating the DB2 Information Center installed on your computer or intranet server	352
DB2 tutorials	354
DB2 troubleshooting information	354
Terms and Conditions	354
Appendix C. Notices	357
Index	361

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 3, “What's changed,” on page 219. This information points out important changes that you need to know before using Version 9.7.

For DB2 Connect information, read Part 2, “DB2 Connect enhancements and changes summary,” on page 207.

This book was updated since it was first published to include details about Fix Pack 1. For a list of important changes included in that fix pack, see Appendix A, “DB2 Version 9.7 for Linux, UNIX, and Windows fix pack summary,” on page 327.

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 21

This chapter describes the new pureXML[®] features and enhancements.

Chapter 5, “Monitoring enhancements,” on page 37

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 57

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 67

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 81

This chapter describes the new features and enhancements that help you port existing database applications from other vendors to DB2 Version 9.7 environments.

Chapter 9, “Workload management enhancements,” on page 87

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 97

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

Chapter 11, “Application development enhancements,” on page 107

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 169

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

Chapter 13, “DB2 Text Search and Net Search Extender enhancements,” on page 179

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

Chapter 14, “Installation, upgrade, and fix pack enhancements,” on page 183

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

Part 2, “DB2 Connect enhancements and changes summary,” on page 207

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 19, “Changed functionality,” on page 221

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 20, “Deprecated functionality in Version 9.7,” on page 275

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 21, “Discontinued functionality in Version 9.7,” on page 305

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

Chapter 22, “Summary of deprecated and discontinued DB2 functionality in Version 9.7 and earlier releases,” on page 313

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, “DB2 Version 9.7 for Linux, UNIX, and Windows fix pack summary,” on page 327

This appendix contains the list of functionality added or changed during Version 9.7 fix packs.

Appendix B, “Overview of the DB2 technical information,” on page 345

This appendix contains information about accessing and using the latest documentation for your DB2 database systems.

Appendix C, “Notices,” on page 357

This appendix contains the legal requirements and limitations related to using the DB2 database product and its documentation.

Highlighting conventions

Topics that are associated with a specific fix pack include an "FPx" prefix at the beginning of the topic title, where *x* represents a fix pack level.

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.
<i>Italics</i>	Indicates one of the following: <ul style="list-style-type: none">• 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: <ul style="list-style-type: none">• 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 21

This chapter describes the new pureXML features and enhancements.

Chapter 5, “Monitoring enhancements,” on page 37

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 57

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 67

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 81

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 87

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 97

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

Chapter 11, “Application development enhancements,” on page 107

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 169

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

Chapter 13, “DB2 Text Search and Net Search Extender enhancements,” on page 179

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

Chapter 14, "Installation, upgrade, and fix pack enhancements," on page 183
This chapter describes the new features and enhancements that make it faster to deploy DB2 products and easier to maintain them.

Part 2, "DB2 Connect enhancements and changes summary," on page 207
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

FP3: Advanced product capabilities

DB2 Advanced Enterprise Server Edition for Linux, UNIX, and Windows is based on the DB2 ESE edition.

DB2 Advanced Enterprise Server Edition is an ideal foundation for building on-demand, enterprise-wide solutions. For example, you can build multi-terabyte databases; high-performance, 24x7-availability, high-volume transaction-processing business solutions; or web-based solutions. This edition includes additional advanced product capabilities in areas such as compression, performance, replication, and security. You can deploy DB2 Advanced Enterprise Server Edition on Linux, UNIX, or Windows servers with any number of CPUs.

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 30)

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: ¹Creation 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:

"Compression dictionary creation" in Database Administration Concepts and Configuration Reference

"Row compression" in Database Administration Concepts and Configuration Reference

Related reference:

"ALTER TABLE " in SQL Reference, Volume 2

"CREATE TABLE " in SQL Reference, Volume 2

"ADMINTABINFO administrative view and ADMIN_GET_TAB_INFO_V97 table function - retrieve table size and state information" in Administrative Routines and Views

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 extract 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” on page 10)
- 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 15)
- Larger data warehouses support (see “Distribution map supports larger data warehouses” on page 15)
- Reduced downtime when adding database partition servers (see “Database partition servers are online immediately after being added to an instance” on page 16)
- Additional index information from the DESCRIBE command (see “DESCRIBE command provides additional index information” on page 17)
- Reorganization of data or indexes for a specific data partition of a data partitioned table (see “FP1: Data partitions and partitioned indexes can be reorganized” on page 17)
- Easier to relocate databases (see “FP1: relocating databases using the db2relocatedb command has been improved” on page 18)
- The ability to restrict database connections within quiesced instances (see “FP2: New **RESTRICTED ACCESS** option restricts database connections within quiesced instance” on page 18)

- The **db2look** command generates DDL statements for database objects and dependents in different schemas (see “FP4: db2look command enhances DDL statement generation for database objects and dependents” on page 19)
- “FP5: Access control, error checking, and table exclusion during data redistribution have been improved” on page 19.

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:

"SNAPSTORAGE_PATHS administrative view and SNAP_GET_STORAGE_PATHS table function - Retrieve automatic storage path information" in Administrative Routines and Views

"SNAPTbsp_PART administrative view and SNAP_GET_TBSP_PART_V91 table function - Retrieve tablespace_nodeinfo logical data group snapshot information" in Administrative Routines and Views

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

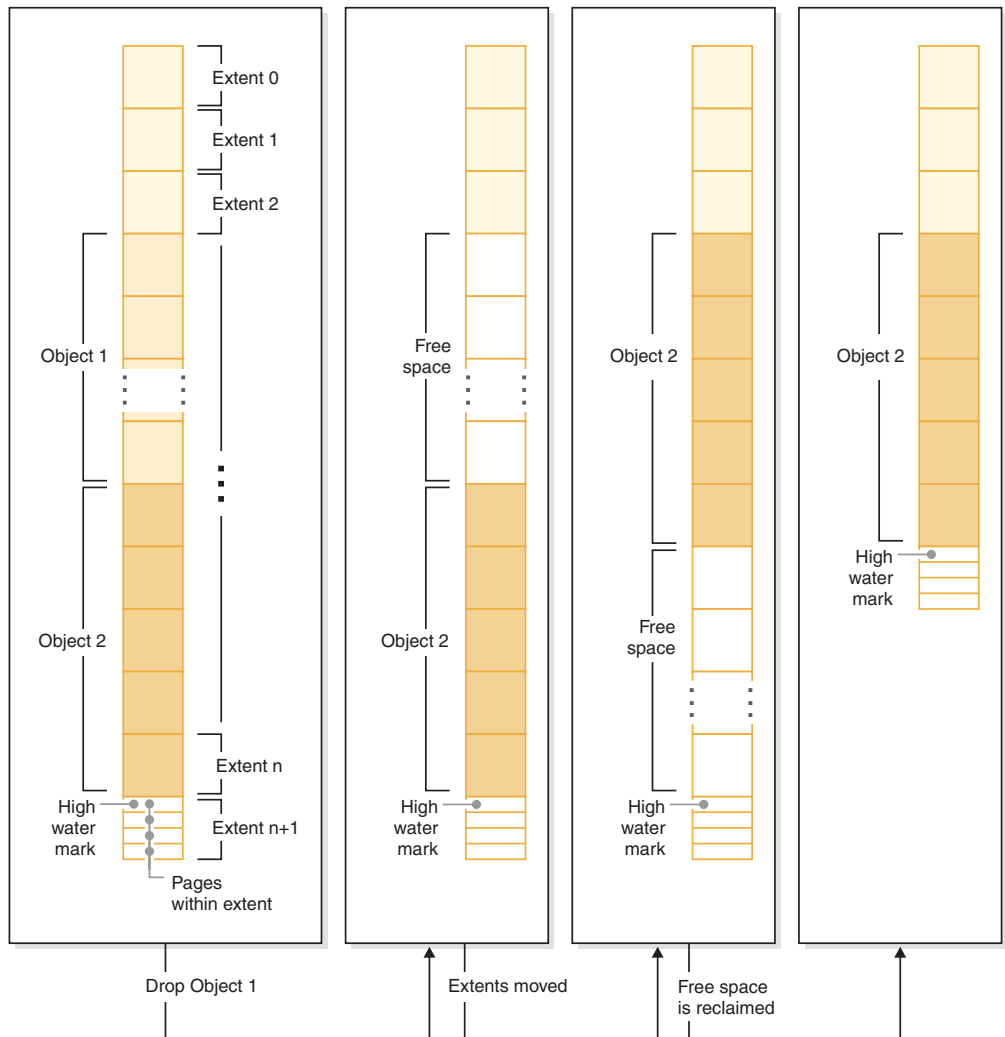
Reclaiming unused space from table spaces is supported

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 toward 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

Related reference:

"AUTOMAINT_SET_POLICY procedure - configure automatic maintenance policy" in Administrative Routines and Views

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 254. 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.

Starting in Version 9.7 Fix Pack 1 and later fix packs, you can access the target table during the copy and swap phases by issuing `NO_TARGET_LOCKSIZE_TABLE` option which disables the default behavior of the locksize table. You can also specify the option that enables to read the data from the source table with or without an `ORDER BY` clause. This option improves the data movement speed.

Related reference:

"ADMIN_MOVE_TABLE procedure - Move tables online" in Administrative Routines and Views

"ADMIN_MOVE_TABLE_UTIL procedure - Modify the online move table procedure" in Administrative Routines and Views

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 293

"sqlugrpn API has been deprecated" on page 293

Related reference:

"SYSCAT.PARTITIONMAPS " in SQL Reference, Volume 1

"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

FP1: Data partitions and partitioned indexes can be reorganized

In Version 9.7 Fix Pack 1 and later fix packs, you can use the **REORG** command on a partitioned table to perform a reorganization of the data of a specific partition or the partitioned indexes of a specific partition. Only access to the specified data partition is restricted, the remaining data partitions of the table retain full read and write access.

On a partitioned table, using the **REORG TABLE** or **REORG INDEXES ALL** command with the **ON DATA PARTITION** clause specifying a partition of the table supports the following features:

- **REORG TABLE** performs a classic table reorganization on the specified data partition while allowing the other data partitions of the table to be fully accessible for read and write operations when there are no nonpartitioned indexes (other than system-generated XML path indexes) on the table. The supported access modes on the partition being reorganized are **ALLOW NO ACCESS** and **ALLOW READ ACCESS**. When there are nonpartitioned indexes on the table (other than system-generated XML path indexes), the **ALLOW NO ACCESS** mode is the default and the only supported access mode for the entire table.
- **REORG INDEXES ALL** performs an index reorganization on a specified data partition while allowing full read and write access to the remaining data partitions of the table. All access modes are supported.

You can issue **REORG TABLE** commands and **REORG INDEXES ALL** commands on a data partitioned table to concurrently reorganize different data partitions or partitioned indexes on a partition. When concurrently reorganizing data partitions or the partitioned indexes on a partition, users can access the unaffected partitions but cannot access the affected partitions. All the following criteria must be met to issue **REORG** commands that operate concurrently on the same table:

- Each REORG command must specify a different partition with the **ON DATA PARTITION** clause.
- Each REORG command must use the ALLOW NO ACCESS mode to restrict access to the data partitions.
- The partitioned table must have only partitioned indexes if issuing **REORG TABLE** commands. No nonpartitioned indexes (except system-generated XML path indexes) can be defined on the table.

The db2Reorg API also supports reorganization of a data partition or its partitioned indexes.

Related concepts:

"Table reorganization" in Troubleshooting and Tuning Database Performance

"Index reorganization" in Troubleshooting and Tuning Database Performance

Related reference:

"REORG INDEXES/TABLE " in Command Reference

FP1: relocating databases using the db2relocatedb command has been improved

Starting in Fix Pack 1, you can specify additional keywords in the **db2relocatedb** command configuration file that make it easier to relocate a database when the paths used are different.

The db2relocatedb configuration file can contain new values for the **mirrorlogpath**, **failarchivepath**, **logarchmeth1**, **logarchmeth2**, and **overflowlogpath** database configuration parameters. When you run the **db2relocatedb** command, the database configuration parameters of the relocated database are updated with the values specified in the configuration file. If you do not specify any of the new keywords, the relocated database maintains the original parameters values.

Related reference:

"db2relocatedb - Relocate database " in Command Reference

FP2: New RESTRICTED ACCESS option restricts database connections within quiesced instance

Starting with DB2 Version 9.7 Fix Pack 2, the new **RESTRICTED ACCESS** option can be specified to prevent authorization checking for all connect attempts to the databases of a quiesced DB2 instance. The new option can also be used when there is a need to have exclusive connections to a database within the quiesced instance.

When the **RESTRICTED ACCESS** option is specified using the **QUIESCE INSTANCE** or **START DATABASE MANAGER** commands, or the db2InstanceQuiesce or db2InstanceStart APIs, authorization checking is prevented to determine if the user ID has DBADM authority. Instance-level authorization checking can still occur; checking a user ID for SYSADM, SYSCTRL, or SYSMANT authority does not require a database to be activated.

With the **RESTRICTED ACCESS** option specified, any user ID trying to connect to a database within a quiesced instance, which has DBADM authority or QUIESCE_CONNECT privilege on the database, will not be allowed to connect. Only user IDs which have SYSADM, SYSCTRL, or SYSMANT authority and the user or group specified with the commands will be allowed to connect to the database.

You can use the **RESTRICTED ACCESS** option when there is a need to have exclusive connections to a database within the quiesced instance. Such cases can include making an offline backup or performing other maintenance activities.

Related reference:

"START DATABASE MANAGER " in Command Reference

"QUIESCE " in Command Reference

"db2InstanceQuiesce - Quiesce instance" in Administrative API Reference

"db2InstanceStart - Start instance" in Administrative API Reference

FP4: db2look command enhances DDL statement generation for database objects and dependents

The **db2look** command, which you can use to help understand, move, or reproduce database objects, has been enhanced to generate DDL statements for parent and dependent objects in different schemas and to generate authorization DDL statements for dependent objects.

Starting with DB2 Version 9.7 Fix Pack 4, you can use a two-part name for a table, in the format of *schema.table*, or a view, in the format of *schema.view*, when the table or view has dependent objects that are in a different schema and you require DDL statements to be also generated for these dependent objects. The ability to specify a two-part name is also extended to selecting tables for DDL statement generation using pattern matching, which you can do by using the **-tw** parameter.

The new **-xdep** and **-xddep** parameters generate authorization DDL statements (for example, GRANT statements) for dependent and parent objects.

Related reference:

"db2look - DB2 statistics and DDL extraction tool " in Command Reference

FP5: Access control, error checking, and table exclusion during data redistribution have been improved

In Version 9.7 Fix Pack 5 and later fix packs, you can specify **QUIESCE DATABASE**, **PRECHECK**, and **EXCLUDE** parameters for the **REDISTRIBUTE DATABASE PARTITION GROUP** command.

Two of the new parameters are applicable only when you perform data redistribution that is not roll-forward recoverable:

- The **PRECHECK** parameter verifies the consistency of the database partition group. The redistribution operation proceeds only if the verification is completed.
- The **QUIESCE DATABASE** command parameter quiesces the database for the duration of the redistribution operation. By quiescing the database, you have greater control over who is allowed to attach or connect to the database during the data redistribution.

By default, when you specify the **NOT ROLLFORWARD RECOVERABLE** parameter, both the **PRECHECK** and **QUIESCE DATABASE** parameters are set to YES.

Specifying the **EXCLUDE** parameter is an alternative to specifying the full list of tables for inclusion in the redistribution operation. You can use this parameter to specify a list of tables to omit from the redistribution operation. For example, you can temporarily omit a table until you can configure it to meet the requirements for data redistribution.

Related concepts:

"Data redistribution" in Partitioning and Clustering Guide

Related reference:

"REDISTRIBUTE DATABASE PARTITION GROUP " in Command Reference

"REDISTRIBUTE DATABASE PARTITION GROUP command using the ADMIN_CMD procedure" in Administrative Routines and Views

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 22)
- Multidimensional clustering (MDC) tables (see “MDC tables support XML columns” on page 23)
- Declared temporary tables (see “Declared temporary tables support XML columns” on page 23)
- User-defined functions (see “Inlined SQL functions support the XML data type” on page 24)
- Partitioned database environments (see “Partitioned database environments support the pureXML feature” on page 25)
- Global variables (see “FP6: XML data type support added in global variables and compiled SQL functions” on page 26)
- Compiled SQL functions (see “FP6: XML data type support added in global variables and compiled SQL functions” on page 26)

The following other enhancements extend pureXML support:

- Indexes over XML data on a partitioned table can be partitioned (see “Partitioned indexes on partitioned tables improve performance” on page 27).
- The DESCRIBE command provides information about system-generated XML indexes (see “DESCRIBE command provides additional index information” on page 17).
- 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 29).
- Predicate pushdown is available for SQL/XML and XQuery statements (see “Predicate pushdown is available for SQL/XML and XQuery statements” on page 30).
- XML parsing and validation can return more detailed messages (see “XML parsing and validation can return more detailed messages” on page 30.)
- 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 30).
- Four new XQuery functions return current date and time values (see “FP1: XQuery functions make it easier to retrieve date and time values for local time zones” on page 31).

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 32).

- Optimization profiles support guidelines for XML data (see “Optimization profiles support guidelines for XML data” on page 32).
- 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 33).
- Distribution statistics are collected for indexes over XML data. (see “FP1: Distribution statistics collected for XML columns” on page 34).

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 127

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 127

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 Partitioning and Clustering Guide

"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 127

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 127

"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 127

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

"XML data is passed by reference in SQL stored procedures" on page 268

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 127
- "Introduction to XQuery" in pureXML Guide

Related reference:

- "CREATE INDEX " in SQL Reference, Volume 2
- "RUNSTATS " in Command Reference
- "Restrictions on the pureXML feature" in pureXML Guide

FP6: XML data type support added in global variables and compiled SQL functions

Starting with Version 9.7 Fix Pack 6, you can create global variables of XML data type, you can specify the XML data type in parameters and the RETURNS clause of compiled SQL functions, and you can define local XML variables in compiled SQL functions.

If you are migrating Oracle applications that use XML variables or XML function parameters, this new support can help make that migration easier.

Depending on the size of the XML documents, using XML global variables and XML data type in compiled SQL functions might require additional space in the system temporary table space. You must ensure that there is enough free space in the system temporary table space.

This new capability is available only in DB2 single partition environments.

The following restrictions apply to XML global variables:

- You cannot specify a default value other than NULL.
- You cannot specify a constant value other than NULL.
- You can assign only well-formed documents to XML global variables.

All existing restrictions for compiled SQL function still apply.

Examples

The following example shows how to create an XML global variable:

```
CREATE OR REPLACE VARIABLE MYSCHEMA.CUSTOMER_HISTORY_VAR XML
```

The following example shows how to specify an input and output XML parameter when creating a compiled SQL function:

```
CREATE FUNCTION update_xml_phone
  (IN   regionNo VARCHAR(8),
   INOUT phone_xml XML)
RETURNS VARCHAR(28)
LANGUAGE SQL
NO EXTERNAL ACTION
BEGIN
  DECLARE tmp_full_phone VARCHAR(28);
  SET tmp_full_phone = regionNo ||
    XMLCAST(XMLQUERY('$p/phone' PASSING phone_xml AS "p") AS VARCHAR(20));
  SET phone_xml = XMLELEMENT (NAME "phone", tmp_full_phone);
  RETURN tmp_full_phone;
END
```

The following example shows how to specify XML in the RETURNS clause when creating a compiled SQL function:

```
CREATE FUNCTION return_phone_number( cid INTEGER)
RETURNS XML
LANGUAGE SQL
NO EXTERNAL ACTION
BEGIN
  RETURN
    SELECT XMLELEMENT (NAME "phone", phone_number) FROM customer WHERE customer_id = cid
END
```

Related concepts:

"Restrictions on SQL functions" in SQL Procedural Languages: Application Enablement and Support

Related reference:

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

"CREATE VARIABLE " in SQL Reference, Volume 2

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 data 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 re-create 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 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.

Starting with DB2 V9.7 Fix Pack 1, when creating a table that uses both multidimensional clustering (MDC) and data partitioning, the system-created MDC block indexes are created as partitioned indexes. Data partitioned MDC tables can take advantage of the features available with partitioned tables such as the rolling in and rolling out of table data. For MDC tables that use table partitioning created with DB2 V9.7 and earlier, the block indexes are nonpartitioned.

Partitioned indexes over XML data

On partitioned tables, indexes over XML data that you create with DB2 V9.7 or earlier are nonpartitioned. Starting in DB2 Version 9.7 Fix Pack 1, you can create an index over XML data on a partitioned table as either partitioned or nonpartitioned. The default is a partitioned index.

To create a nonpartitioned index, specify the NOT PARTITIONED option for the CREATE INDEX statement. To convert a nonpartitioned index over XML data to a partitioned index:

1. Drop the nonpartitioned index.
2. Create index by using the CREATE INDEX statement without the NOT PARTITIONED option.

Related concepts:

"Partitioned tables" in Partitioning and Clustering Guide

"Optimization strategies for partitioned tables" in Partitioning and Clustering Guide

"Table partitioning and multidimensional clustering tables" in Partitioning and Clustering Guide

"Block indexes for MDC tables" in Partitioning and Clustering Guide

Related tasks:

"Converting existing indexes to partitioned indexes" in Partitioning and Clustering Guide

Related reference:

"ALTER TABLE " in SQL Reference, Volume 2

"CREATE INDEX " in SQL Reference, Volume 2

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

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

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 127

"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*

"ADMIN_IS_INLINED function - Determine if data is inlined" in *Administrative Routines and Views*

"ADMIN_EST_INLINE_LENGTH function - Estimate length required to inline data" in *Administrative Routines and Views*

FP1: XQuery functions make it easier to retrieve date and time values for local time zones

Four new XQuery functions return current date and time values using the local time zone of the DB2 database system.

The functions are `db2-fn:current-local-time()`, `db2-fn:current-local-date()`, `db2-fn:current-local-dateTime()`, and `db2-fn:local-timezone()`. They differ from `fn:current-time()`, `fn:current-date()`, and `fn:current-dateTime()`, which return the date

and time values in the Coordinated Universal Time (UTC) implicit timezone and include a time zone component in the returned value.

For example, if the `fn:current-time()` function was invoked on November 20, 2009 at 13:00 on a DB2 database system in Toronto (timezone `-PT5H`), the returned value might be `18:00:50.282691Z`, whereas the `db2-fn:current-local-time()` function would return the value `13:00:50.282691`.

Related reference:

- "current-local-time function" in XQuery Reference
- "current-local-date function" in XQuery Reference
- "current-local-dateTime function" in XQuery Reference
- "local-timezone function" in XQuery Reference

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 about 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

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 127

Related reference:

"CREATE INDEX " in SQL Reference, Volume 2

"REORG INDEXES/TABLE " in Command Reference

FP1: Distribution statistics collected for XML columns

Starting with DB2 Version 9.7 Fix Pack 1, distribution statistics can be collected for XML columns to support faster queries over the data in XML columns. Distribution statistics are collected for indexes over XML data of type VARCHAR, DOUBLE, TIMESTAMP, and DATE.

For DB2 V9.7 Fix Pack 1 and later, distribution statistics on an XML column can be collected.

- Distribution statistics are collected for indexes over XML data of type VARCHAR, DOUBLE, TIMESTAMP, and DATE. XML distribution statistics are not collected for indexes over XML data of type VARCHAR HASHED.
- Distribution statistics are collected for each index over XML data specified on an XML column.
- XML distribution statistics are collected when automatic table RUNSTATS operations are performed.

To collect distribution statistics on an XML column using the RUNSTATS utility, both distribution statistics and table statistics must be collected. Table statistics must be gathered in order for distribution statistics to be collected because XML distribution statistics are stored with table statistics. As the default, the RUNSTATS utility collects a maximum of 250 quantiles for distribution statistics for each index over XML data. The maximum number of quantiles for a column can be specified when executing the RUNSTATS utility.

The following list describes the situations in which XML distribution statistics are not created or collected:

- XML distribution statistics are not created when loading data with the STATISTICS option.
- XML distribution statistics are not collected for partitioned indexes over XML data defined on a data partitioned table.
- XML distribution statistics are not collected when collecting only index statistics, or collecting index statistics during index creation.

Related concepts:

"Catalog statistics" in Troubleshooting and Tuning Database Performance

"Guidelines for collecting and updating statistics" in Troubleshooting and Tuning Database Performance

Related tasks:

"Collecting distribution statistics for specific columns" in Troubleshooting and Tuning Database Performance

Related reference:

"RUNSTATS " in Command Reference

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
REDISTRIBUTE DATABASE PARTITION GROUP command		Creates a compression dictionary ¹

Note: ¹Creation 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:

"Compression dictionary creation" in Database Administration Concepts and Configuration Reference

"Row compression" in Database Administration Concepts and Configuration Reference

Related reference:

"ALTER TABLE " in SQL Reference, Volume 2

"CREATE TABLE " in SQL Reference, Volume 2

"ADMINTABINFO administrative view and ADMIN_GET_TAB_INFO_V97 table function - retrieve table size and state information" in Administrative Routines and Views

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” on page 38)
- The maximum number of active event monitors has been increased (see “Maximum number of active event monitors is increased” on page 39)
- New relational monitoring interfaces for locking events (see “FP1: New relational monitoring interfaces for locking events” on page 40)
- A high granularity of control over what information to collect (see “New monitor elements and database configuration parameters provide more granular monitoring” on page 41)
- 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 42)
- A package cache event monitor to capture information about dynamic and static SQL statement entries after having been flushed from the database package cache (see “FP1: New event monitor for dynamic and static SQL statements in package cache” on page 48)
- New lock event monitors to simplify problem determination for deadlocks, lock timeout, and lock waits (see “Lock event reporting has been enhanced” on page 46)
- Improved transaction monitoring of unit of work and total CPU usage (see “A new unit of work event monitor supports transaction monitoring” on page 43)
- The date that an object was last used can now be determined to help manage objects (see “FP1: Last referenced date is available for tables, table partitions, indexes, and packages” on page 45)

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

- Monitor elements indicating where and how the DB2 database manager spends its waiting time (see “Time-spent monitor elements are more comprehensive” on page 44)
- Viewing system metrics from the statistics event monitor without having to parse an XML document, and without having to manually calculate changes in value

- Statistics event monitor produces XML document metrics that reports system metrics for most recent monitoring interval
- Statistics event monitor produces XML document metrics that reports system metrics for most recent monitoring interval
- Information reporting about statements in the package cache, including static and dynamic statements (see “New relational monitoring interfaces are light weight and SQL accessible”)
- Capture of information about cached statement entries after they have been flushed from the database package cache (see “FP1: New event monitor for dynamic and static SQL statements in package cache” on page 48)
- 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 46)
- The section explain functionality captures explain information about a statement using only the contents of the runtime section (see “FP1: Statements from a runtime section can be explained” on page 47)
- The explain facility is enhanced with actual values for operator cardinality (see “FP1: Explain enhanced with actual values for operator cardinality” on page 48)
- Progress information about the **RUNSTATS** command, table and index reorganization as well as additional system monitoring information can be generated (see “Additional system monitoring information can be generated” on page 49)
- Monitoring table functions information can be viewed using administrative views (see “FP1: Monitoring table functions information can be viewed using administrative views” on page 50)
- Table functions for row-based formatting of monitoring information (see “FP1: Table functions for row-based formatting of monitoring information are available” on page 51)
- Table functions for FCM monitoring (see “FP2: Identify FCM issues more easily” on page 52)
- Ability to view information about memory usage, and about the underlying computer system and network using table functions. See , and “FP6: New table functions provide SQL-based access to system information” on page 54 for more information.

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
- MON_GET_PKG_CACHE_STMT_DETAILS (Only available starting with DB2 Version 9.7 FixPack 1.)

Data object level

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

Maximum number of active event monitors is increased

In previous versions of the DB2 database manager, the maximum number of active event monitors has been limited to 32 per database partition. In addition, you could have only one active workload management (WLM) event monitor from each of the activities, statistics, and threshold violations types. In DB2 Version 9.7, the permitted maximum number of active event monitors has been increased.

The following list contains the details of the new increased maximum number limits of active event monitors:

- Although an unlimited number of event monitors may be defined, a maximum of 128 event monitors can be active simultaneously on each database partition.
- In a partitioned database environment, a maximum of 32 GLOBAL event monitors can be active simultaneously on each database.

Note: Only deadlock file and pipe event monitors can be global in scope. However, the deadlock event monitors have been deprecated. For more details, see: “CREATE EVENT MONITOR FOR DEADLOCKS statement and DB2DETAILDEADLOCK event monitor have been deprecated”.

- Multiple event monitors, for any of the activities, statistics, or threshold violations types, can now be active on any database partition.

Related concepts:

"CREATE EVENT MONITOR FOR DEADLOCKS statement and DB2DETAILDEADLOCK event monitor have been deprecated" on page 298

Related tasks:

"Collecting workload management statistics using a statistics event monitor" in Workload Manager Guide and Reference

"Monitoring threshold violations" in Workload Manager Guide and Reference

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

FP1: New relational monitoring interfaces for locking events

Starting with Version 9.7 Fix Pack 1, the MON_GET_APPL_LOCKWAITS, MON_GET_LOCKS, and MON_FORMAT_LOCK_NAME relational monitoring interfaces can be used to collect locking event data to help you quickly identify locking issues that might be causing problems.

The monitoring interfaces are more efficient and have a lower impact on the system than existing snapshot interfaces. These new interfaces report monitoring elements related to locking events. Use the following routines to collect information about locks:

- MON_GET_APPL_LOCKWAITS
 - Returns information about the locks that all the applications are waiting to acquire on the currently connected database.
- MON_GET_LOCKS
 - Returns a list of all locks on the currently connected database.
- MON_FORMAT_LOCK_NAME
 - Formats the internal lock name and returns details about the lock in a row-based format. Each row consists of a key-value pair pertaining to a particular lock.

Use the following administrative view to collect lock wait information:

- MON_LOCKWAITS
 - Returns information about agents working on behalf of applications that are waiting to obtain locks in the currently connected database. It is a useful query for identifying locking problems.

The new lock-related monitoring interfaces replace the following deprecated administrative views and table functions:

- SNAPLOCK administrative view and SNAP_GET_LOCK table function
- SNAPLOCKWAIT administrative view and SNAP_GET_LOCKWAIT table function
- LOCKS_HELD administrative view
- LOCKWAITS administrative view

Related reference:

"MON_LOCKWAITS administrative view - Retrieve metrics for applications that are waiting to obtain locks" in Administrative Routines and Views

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 38.

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.
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 routines and views" in Administrative Routines and Views

"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).

Related reference:

"MON_GET_PKG_CACHE_STMT table function - Get SQL statement activity metrics in the package cache" in Administrative Routines and Views

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.

In Version 9.7 Fix Pack 1 and later fix packs, the unit of work event monitor can also collect a listing of packages used within each unit of work, including the nesting level and the elapsed time for each package. Unique information is collected for each invocation of a routine. The package listing information helps facilitate stored procedure troubleshooting.

After the unit of work event monitor data has been captured, you can access it using one of the following methods:

- An XML document created by the new EVMON_FORMAT_UE_TO_XML table function
- Relational tables populated by the new EVMON_FORMAT_UE_TO_TABLES procedure
- An XML or text document using the 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

"MON_GET_UNIT_OF_WORK table function - Get unit of work metrics" in Administrative Routines and Views

"MON_GET_UNIT_OF_WORK_DETAILS table function - Get detailed unit of work metrics" in Administrative Routines and Views

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

"EVMON_FORMAT_UE_TO_XML table function - convert unformatted events to XML" in Administrative Routines and Views

"EVMON_FORMAT_UE_TO_TABLES procedure - move an XML document to relational tables" in Administrative Routines and Views

"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.

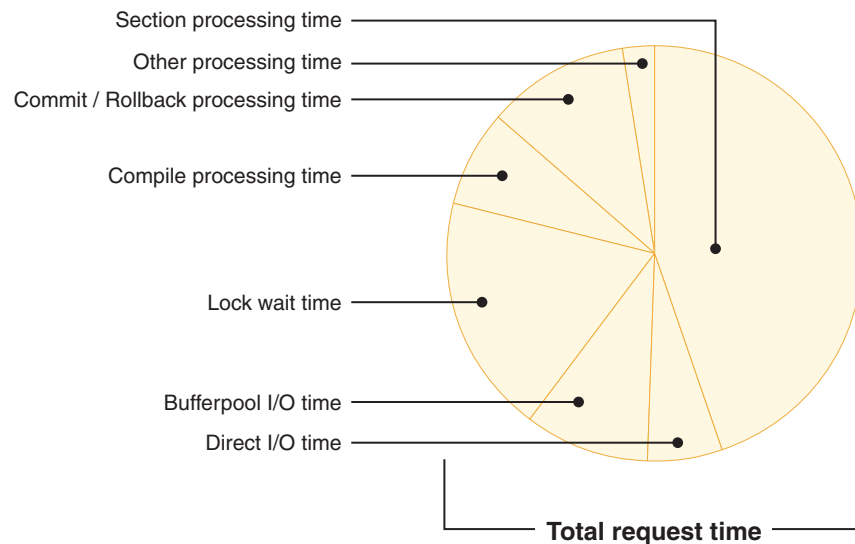
The new time-spent monitor elements, including wait times and component times, provide the following information:

- 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.
- Starting in DB2 Version 9.7 Fix Pack 1, detailed breakdown of processing time by component (such as compilation, or section execution). This breakdown allows you to identify the primary contributors to processing 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 following diagram shows one possible visualization of the total DB2 request time on a particular system:



In this example, the section processing time is responsible for a significant percentage of the total request time. This is generally desirable because section processing time represents the time that is spent doing core SQL processing rather than waiting on resources or driving transaction end processing. On the other hand, a large percentage of the overall request time is also spent in various waiting situations; lock wait time specifically. This percentage of lock wait time is undesirable and indicates the need to investigate the locking behavior in more detail.

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" *What's New for DB2 Version 9.7*.

Related concepts:

"Time-spent monitor elements" in Database Monitoring Guide and Reference

FP1: Last referenced date is available for tables, table partitions, indexes, and packages

The last referenced date indicates the last date that an object was used and is available for tables, table data partitions, indexes, packages, and materialized query tables (MQTs). The last referenced date is accessed through the LASTUSED column of the corresponding catalog table for the object.

The last referenced date is used to identify objects that have not been accessed for an extended period of time and might be considered candidates for removal. For example, removing indexes that are never used in queries saves both disk space and maintenance overhead (that is, overhead when insertions and updates are performed on the table on which the index was defined).

Related concepts:

"Determining the date a database object was last used" in Database Monitoring Guide and Reference

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 the event monitor data has been captured, the following is a list of the methods you can use to access that data:

- An XML document created by the new EVMON_FORMAT_UE_TO_XML table function
- Relational tables populated by the new EVMON_FORMAT_UE_TO_TABLES procedure
- An XML or text document using the Java-based **db2evmonfmt** tool

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 Troubleshooting and Tuning Database Performance

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

"EVMON_FORMAT_UE_TO_XML table function - convert unformatted events to XML" in Administrative Routines and Views

"EVMON_FORMAT_UE_TO_TABLES procedure - move an XML document to relational tables" in Administrative Routines and Views

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 **db2exp1n** 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 concepts:

"Guidelines for capturing section explain information" in Troubleshooting and Tuning Database Performance

Related tasks:

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

Related reference:

"db2expln - SQL and XQuery Explain " in Command Reference

FP1: Statements from a runtime section can be explained

Starting with Version 9.7 Fix Pack 1, DB2 database manager has the ability to perform an explain directly from the contents of a runtime section. This functionality is known as a section explain. The advantage of a section explain over a traditional explain using an EXPLAIN statement is that the EXPLAIN statement will recompile the statement that is being explained. If the compilation environment or table statistics have changed when the EXPLAIN statement is issued, the compiler may generate a different access plan.

A section explain will always provide the exact access plan that was executed, since the access plan is reconstructed directly from the executable section. A section explain is similar to the functionality provided by the **db2expln** command, but provides a level of detail approaching that which is provided by the EXPLAIN statement.

The section explain functionality is accessible through a set of stored procedures. The stored procedures take input parameters that are used to locate a section (either in memory, catalogs, captured by an event monitor, or provided directly as input) and perform the explain, populating the explain tables similar to the EXPLAIN statement. The stored procedures output the key fields for the explain instance that was populated in the explain tables. These key fields can be used as input to existing explain formatting tools, for example **db2exfmt**, which extract the information from the explain tables and present it in a formatted output.

The section explain procedures are the following:

- EXPLAIN_FROM_ACTIVITY
- EXPLAIN_FROM_CATALOG
- EXPLAIN_FROM_DATA
- EXPLAIN_FROM_SECTION

Related concepts:

"Guidelines for capturing section explain information" in Troubleshooting and Tuning Database Performance

FP1: Explain enhanced with actual values for operator cardinality

Starting with Version 9.7 Fix Pack 1, runtime statistics can be gathered for access plan operators during the execution of a section. These statistics are referred to as section actuals. In Fix Pack 1, the only statistic available is cardinality for access plan operators.

The explain facility output conveniently displays both the section actuals and estimated access plan values for your comparison. The result of this comparison can point to out-of-date statistics used by the optimizer to select an incorrect access plan. Action can then be taken to update the statistics using the **RUNSTATS** command and then retrying the application with an up-to-date access plan in place. Section actuals are only available when a section explain is performed and the section was captured using an activity event monitor.

Note:

- Section actuals must be enabled (set to BASE) using the **section_actuals** database configuration parameter or for a specific application using the **WLM_SET_CONN_ENV** procedure. Section actuals cannot be enabled if automatic statistics profile generation (**auto_stats_prof**) is enabled in the database configuration (SQLCODE -5153).
- The **section_actuals** setting specified by the **WLM_SET_CONN_ENV** procedure for an application takes effect immediately. Section actuals will be collected for the next statement issued by the application.

The ability to collect section actuals information can help to resolve SQL query performance slow downs.

Related concepts:

"Capturing and accessing section actuals" in Troubleshooting and Tuning Database Performance

Related reference:

"section_actuals - Section actuals configuration parameter" in Database Administration Concepts and Configuration Reference

FP1: New event monitor for dynamic and static SQL statements in package cache

Starting with Version 9.7 Fix Pack 1, the package cache event monitor (CREATE EVENT MONITOR FOR PACKAGE CACHE) records events from both dynamic and static SQL statements when they are flushed from the database package cache.

The new package cache event monitor captures information about cached statement entries after they have been flushed from the database package cache. The event monitor captures an accurate history about statements, that were in the package cache, which can help to resolve SQL query performance and problem determination issues.

The core data collected for a package cache event are the monitor elements reported through the MON_GET_PKG_CACHE_STMT table function. In addition,

the event monitor collects information about the executable section of the activity. The collected information is the same for both dynamic and static SQL statements.

After the event monitor data has been captured, the following is a list of the methods you can use to access that data:

- An XML document created by the new `EVMON_FORMAT_UE_TO_XML` table function
- Relational tables populated by the new `EVMON_FORMAT_UE_TO_TABLES` procedure
- An XML or text document using the Java-based **db2evmonfmt** tool

Related concepts:

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

Related tasks:

"Collecting package cache event data and generating reports" in Database Monitoring Guide and Reference

Related reference:

"CREATE EVENT MONITOR (package cache) statement" in SQL Reference, Volume 2

"EVMON_FORMAT_UE_TO_XML table function - convert unformatted events to XML" in Administrative Routines and Views

"EVMON_FORMAT_UE_TO_TABLES procedure - move an XML document to relational tables" in Administrative Routines and Views

"MON_GET_PKG_CACHE_STMT table function - Get SQL statement activity metrics in the package cache" in Administrative Routines and Views

"MON_GET_PKG_CACHE_STMT_DETAILS - get detailed metrics for package cache entries" in Administrative Routines and Views

Additional system monitoring information can be generated

Version 9.7 contains **db2pd** command improvements that make it easier to monitor system activities, including table and index reorganization progress information.

Starting in Version 9.7, a summary option is available for the **-pages** command parameter, and new command parameters (**-wlocks** and **-apinfo**) are also available.

You can use the summary option for the **-pages** parameter to generate a more compact report that contains only the buffer pool summary information section. Additional columns that include information about table space IDs, dirty pages, permanent pages, and temporary pages are displayed in the summary section.

You can use the **-wlocks** parameter to monitor dynamically the applications with locks that are in lock wait mode. You can use the **-apinfo** parameter to capture detailed runtime information about a specific application or for all applications. Both parameters have options to save the information into separate files.

In Version 9.7 Fix Pack 1 and later fix packs, you can use the `index` option of the **-reorgs** parameter to display progress information about index reorganizations. Also, you can use the **-runstats** parameter to display progress information about table and index **RUNSTATS** operations. Starting in Version 9.7 Fix pack 2, you can also use the same option to monitor index reorganizations for partitioned indexes at the table and partition levels.

The result of the monitoring is reported in separate output for partitioned and non-partitioned indexes. Each partition will have its index reorganization reported in its own output.

Related concepts:

"Monitoring and troubleshooting using db2pd command" in Troubleshooting and Tuning Database Performance

Related reference:

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

FP1: Text reports can be generated based on monitoring data

Starting in Version 9.7 Fix Pack 1, you can use the procedures in the new MONREPORT procedure module to retrieve monitoring information and generate easy-to-read reports.

The monitoring reports contain information reported by the monitoring table functions. They also include metrics (such as ratios and percentages) calculated at specified intervals.

The following reports are currently available:

- The CURRENTAPPS procedure generates a report about the current state of processing of units or work, agents, and activities for each connection.
- The CURRENTSQL procedure generates a report that summarizes currently running activities.
- The CONNECTION procedure generates a report containing monitor data for each connection.
- The DBSUMMARY procedure generates a report that summarizes system and application performance information.
- The LOCKWAIT procedure generates a report that contains information about each lock wait currently in progress. Details include information about the lock holder and requester and characteristics of the lock held and the lock requested.
- The PKGCACHE procedure generates a report that lists the top statements accumulated in the package cache.

You can use these procedures as provided, or you can edit these procedures to customize the reports for your needs.

FP1: Monitoring table functions information can be viewed using administrative views

New administrative views encapsulate key queries using the new monitoring table functions introduced in DB2 Version 9.7 and Version 9.7 Fix Pack 1.

The new monitoring table functions introduced in DB2 Version 9.7 and Version 9.7 Fix Pack 1 provide many detailed metrics describing the database objects and environment. To see the most important metrics in an easily readable format, you can use the new monitoring administrative views. You can simply issue a SELECT * command to see the main metrics from each table function, as well as some common calculated values.

The following administrative views are available:

- MON_BP_UTILIZATION

- MON_TBSP_UTILIZATION
- MON_LOCKWAITS
- MON_PKG_CACHE_SUMMARY
- MON_CURRENT_SQL
- MON_CURRENT_UOW
- MON_SERVICE_SUBCLASS_SUMMARY
- MON_WORKLOAD_SUMMARY
- MON_CONNECTION_SUMMARY
- MON_DB_SUMMARY

FP1: Table functions for row-based formatting of monitoring information are available

In Version 9.7 Fix Pack 1 and later fix packs, monitor elements reported in XML documents can be displayed and analyzed in a generic fashion using new row-based formatting table functions.

Detailed monitoring table functions, such as MON_GET_WORKLOAD_DETAILS, return an XML document called DETAILS, containing a number of detailed monitor elements. In addition, the statistics event monitor returns a DETAILS XML document, the activity event monitor returns a DETAILS_XML XML document, the EVMON_FORMAT_UE_TO_XML table function returns an XMLREPORT XML document, and the EVMON_FORMAT_UE_TO_TABLES procedure returns a METRICS XML document. You can review and analyze the monitor elements returned in these XML document by using the new row-based formatting table functions. Which monitor elements are returned depends on the table function or event monitor which produced the XML document.

Important: Starting with Version 9.7 Fix Pack 6, the XML document details_xml is deprecated in the statistics event monitor, and might be removed in a future release. For more information, see Reporting of metrics in details_xml by the statistics event monitor has been deprecated“Reporting of metrics in details_xml by the statistics event monitor has been deprecated” in *What’s New for DB2 Version 9.7*.

MON_FORMAT_XML_WAIT_TIMES_BY_ROW

Returns a list of wait time monitor elements, such as **total_wait_time** and **lock_wait_time** for each XML document.

MON_FORMAT_XML_COMPONENT_TIMES_BY_ROW

Returns a list of component time monitor elements, including processing time monitor elements, such as **total_compile_time**, **total_compile_proc_time**, and **stmt_exec_time**, for each XML document.

MON_FORMAT_XML_TIMES_BY_ROW

Returns formatted row-based output for the combined hierarchy of wait and processing times that are contained in an XML metrics document.

MON_FORMAT_XML_METRICS_BY_ROW

Returns all the metrics contained in the XML document.

FP2: Identify FCM issues more easily

Starting with Version 9.7 Fix Pack 2, the `MON_GET_FCM` and `MON_GET_FCM_CONNECTION_LIST` table functions can be used to collect fast communication manager (FCM) data to help you identify communication issues more easily.

These table functions are more efficient and have a lower impact on the system than existing snapshot interfaces. Use the following table functions to collect information about FCM:

- `MON_GET_FCM`
 - Returns metrics for FCM.
- `MON_GET_FCM_CONNECTION_LIST`
 - Returns monitor metrics for all the FCM connections on the specified member.

In addition, FCM-related metrics have been added to the outputs of both the `db2pd` command and the `GET SNAPSHOT` command.

Related reference:

"GET SNAPSHOT " in Command Reference

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

"MON_GET_FCM - Get FCM metrics" in Administrative Routines and Views

"MON_GET_FCM_CONNECTION_LIST - Get details for all FCM connections" in Administrative Routines and Views

FP3: Simplify capture of detailed statement information using the new db2caem tool with db2support integration options

Starting with V9.7 FP3, a new tool `db2caem` (DB2 Capture Activity Event Monitor data tool) has been created to simplify the process of capturing detailed diagnostic and runtime information about one or more statements. New **db2support** options have been added for the optimizer mode to collect the data captured by **db2caem**.

The `db2caem` tool will cleanup any objects and tables that it creates, this includes the activity event monitor that it creates to perform the capture of statement information. If a you have already captured information for a statement using an existing activity event monitor, you can bypass the capture step by providing as input an activity event monitor name along with an application ID, unit-of-work ID, and activity ID to identify the statement of interest and use the `db2caem` tool to export and format the information only.

db2support integration

New **db2support** event monitor options simplify capturing activity event monitor data with the `db2caem` tool (`-aem`, `-actevm`, `-appid`, `-uowid`, and `-actid` options). These options can be specified by themselves, or can be combined with one of the available SQL statement options to capture data for an SQL statement (`-st`, `-sf` or `-se` option).

Related reference:

"db2support - Problem analysis and environment collection tool " in Command Reference

"db2caem - Capture activity event monitor data tool " in Command Reference

FP4: Simplified access to activity metrics in the activity event monitor

Starting with Version 9.7 Fix Pack 4, activity metrics (including CPU usage details, various counters, and time spent metrics) are now reported individually by the activity event monitor in the new event_activitymetrics logical data group and can now be collected in a relational table.

By default, activity metrics are stored as an XML document in the DETAILS_XML column and individually as columns in the table generated by the event_activitymetrics logical data group. The default name for this table is ACTIVITYMETRICS. In previous releases, activity metrics were collected by the activity event monitor and stored as an XML document in the DETAILS_XML column of the table generated by the activity logical data group.

Start using SQL to access activity metrics. You can access activity metrics using a simple relational query against the event_activitymetrics logical data group, without having to parse or understand the contents of the metrics document available in the DETAILS_XML column. For example, after creating a write to table activity event monitor named A, you can access the **pool_read_time** and **total_cpu_time** elements using a simple SQL statement such as the following:

```
SELECT
pool_read_time,
total_cpu_time
FROM ACTIVITYMETRICS_A as A;
```

The DETAILS_XML element continues to store an XML document containing all the activity metrics, for those users that prefer the XML representation or are using one of the row based metrics formatting table functions to view the metrics in a hierarchical format.

If you only want to use the activity metrics only in a table, remove the DETAILS_XML column by performing the following actions:

```
CREATE EVENT MONITOR event-monitor-name FOR ACTIVITIES WRITE TO TABLE;
ALTER TABLE ACTIVITY_event-monitor-name DROP DETAILS_XML;
REORG TABLE ACTIVITY_event-monitor-name
```

If you want to use the activity metrics only as an XML document, create the activity event monitor by issuing the following statement:

```
CREATE EVENT MONITOR event-monitor-name FOR ACTIVITIES WRITE TO TABLE CONTROL, ACTIVITY,
ACTIVITYSTMT, ACTIVITYVALS
```

Related reference:

"pool_read_time - Total buffer pool physical read time monitor element" in Database Monitoring Guide and Reference

"total_cpu_time - Total CPU time monitor element" in Database Monitoring Guide and Reference

FP5: New monitoring table functions return information about memory

In Version 9.7 Fix Pack 5 and later fix packs, the MON_GET_MEMORY_POOL and MON_GET_MEMORY_SET monitoring table functions provide information about memory usage that was previously available only through snapshot monitoring views and routines.

With the introduction of these two new table functions, the following interfaces are deprecated:

- The SNAP_GET_DBM_MEMORY_POOL table function and SNAPDBM_MEMORY_POOL administrative view
- The SNAP_GET_DB_MEMORY_POOL table function and SNAPDB_MEMORY_POOL administrative view
- The SNAP_GET_AGENT_MEMORY_POOL table function and SNAPAGENT_MEMORY_POOL administrative view

Related reference:

"MON_GET_MEMORY_SET - get memory set information " in Administrative Routines and Views

"MON_GET_MEMORY_POOL - get memory pool information" in Administrative Routines and Views

FP6: New table functions provide SQL-based access to system information

Starting with Version 9.7 Fix Pack 6, you can use new table functions to retrieve information about the system on which the DB2 product runs using SQL queries. This information was previously available only through snapshot monitoring interfaces or administrative views.

Three new table functions provide information about the environment in which the DB2 product runs.

ENV_GET_SYSTEM_RESOURCES

The ENV_GET_SYSTEM_RESOURCES table function returns operating system, CPU, memory, and other information that is related to members on the system.

ENV_GET_DB2_SYSTEM_RESOURCES

The ENV_GET_DB2_SYSTEM_RESOURCES table function returns CPU usage and DB2 process information for specified members in the current instance.

ENV_GET_NETWORK_RESOURCES

The ENV_GET_NETWORK_RESOURCES table function returns information for all active network adaptors on the host machines running the DB2 product.

Use these functions instead of the table function ENV_GET__SYS_RESOURCES, and the administrative view ENV_SYS_RESOURCES, which is deprecated as of Version 9.7 Fix Pack 6.

Related reference:

"Deprecated SQL administrative routines and their replacement routines or views" in Administrative Routines and Views

"ENV_GET_SYSTEM_RESOURCES table function - Return system information" in Administrative Routines and Views

"ENV_GET_NETWORK_RESOURCES table function - Return network information" in Administrative Routines and Views

"ENV_GET_DB2_SYSTEM_RESOURCES table function - Return DB2 system information" in Administrative Routines and Views

FP6: New logical data groups added to the statistics event monitor

Starting with Version 9.7 Fix Pack 6, two new logical data groups of monitor elements have been added to support direct access to individual monitor elements that were previously returned only in XML documents.

The logical data groups event_scmetrics and event_wlmetrics contain monitor elements that report metrics-related information. Previously to Fix Pack 6, metrics information was collected by the statistics event monitor and stored as an XML document called details_xml. This document was included in the event_scmstats and event_wlstats logical data groups. You can now look at the information contained in these monitor elements by viewing them directly. For example, for statistics event monitors that write to tables, you can access the metrics by using an SQL query to retrieve data from a table, instead of having to post-process or parse the XML document contained in the DETAILS_XML column.

The new logical data groups are included by default in the output of the statistics event monitor.

Tip: Unlike the metrics recorded in the details_xml document, the metrics elements reported in these logical data groups show the change in value for the monitor element since the last time statistics were collected.

The details_xml element continues to store an XML document containing all of the system metrics. However, the collection of metrics elements in details_xml is deprecated. If you prefer to use the XML representation of metrics elements, use the new metrics document instead. See "FP6: Reporting of metrics in details_xml by the statistics event monitor has been deprecated" on page 303 and "FP6: XML document metrics stores system metrics collected by statistics event monitor" on page 56 for more information.

Related concepts:

"Capturing system metrics using the statistics event monitor" in Database Monitoring Guide and Reference

Related reference:

"Event monitor logical data groups and monitor elements" in Database Monitoring Guide and Reference

"CREATE EVENT MONITOR (statistics) " in SQL Reference, Volume 2

FP6: XML document metrics stores system metrics collected by statistics event monitor

Starting in Version 9.7 Fix Pack 6, a new XML document called `metrics` collects system metrics monitor elements. Unlike the metrics collected in the `details_xml` document, metrics in the new `metrics` document show the change in value for the monitor element since the last time statistics were collected.

In previous releases, system metrics were captured by the statistics event monitor only in the XML document `details_xml`. This document was generated for both the `event_scstats` and `event_wlstats` logical data groups. The new `metrics` document collects the same metrics monitor elements as are collected in `details_xml`. However, unlike the metrics in `details_xml`, which continue to increase until the next database activation, the metrics collected in the `metrics` document are calculated to show the change in value for the monitor element since the last time statistics were collected.

Note: The collection of metrics elements in `details_xml` is deprecated. See "FP6: Reporting of metrics in `details_xml` by the statistics event monitor has been deprecated" on page 303 for more information. If you prefer to use the XML representation of metrics elements, use the new `metrics` document instead. In addition to viewing system metrics from the `metrics` document, you can view them directly from the output associated with two new logical data groups that have been added for Version 9.7 Fix Pack 6. See "FP6: New logical data groups added to the statistics event monitor" on page 55 for more information.

Related concepts:

"Capturing system metrics using the statistics event monitor" in Database Monitoring Guide and Reference

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

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

The following enhancements are included:

- Read operations support on High Availability and Disaster Recovery (HADR) standby databases (see “FP1: Read operations on HADR standby databases are supported” on page 59)
- The super asynchronous (SUPERASYNC) HADR synchronization mode which ensures that transactions can never be blocked or experience elongated response times due to network interruptions or congestion. (see “FP5: Super asynchronous HADR synchronization mode has been added” on page 64)
- New scripts that enable you to automate DB2 HADR failover operations on Windows operating systems (see “FP4: New scripts improve integration between DB2 High Availability Disaster Recovery and IBM Tivoli System Automation for Multiplatforms (Windows)” on page 62)
- Roving high availability (HA) failover support (see “FP4: Roving high availability (HA) failover reduces downtime in partitioned database environments” on page 63)
- Integrated high availability features for configuring clustered environments on Solaris SPARC (see “Cluster management software integration support has been extended (Solaris)” on page 58)
- DB2 Advanced Copy Services (ACS) support on the AIX® 6.1 operating system (see “FP1: DB2 Advanced Copy Services (ACS) is supported on AIX 6.1” on page 58)
- Improved support for data deduplication devices in backup operations (see “FP3: Data deduplication device support has been integrated into the backup utilities” on page 60)
- Proxy nodes support (see “FP2: Proxy node support for the db2adutl command has been added” on page 60)
- The ability to transport table spaces and SQL schemas (see “FP2: Databases can be restored using transportable sets” on page 61)
- Automatic recovery utilities now recognize merged backups (see “FP5: Automatic recovery utilities now recognize merged backups” on page 63)
- db2adutl enhancements simplify backup image and log file management (see “FP5: db2adutl enhancements simplify backup image and log file management” on page 63)

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” on page 58)

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

FP1: DB2 Advanced Copy Services (ACS) is supported on AIX 6.1

Starting in Version 9.7 Fix Pack 1, DB2 Advanced Copy Services (ACS) for AIX supports the AIX 6.1 operating system.

In previous releases, only AIX 5.3 was supported with DB2 ACS on AIX, requiring users of AIX 6.1 to perform a manual copy or to install the full version of Tivoli Storage Manager for Advanced Copy Services V6.1 for AIX 6.1.

In Version 9.7 Fix Pack 1 and later fix packs, the ACS version that is bundled with the DB2 server product can be used with both AIX 5.3 and 6.1 to take snapshot backups.

Related concepts:

"DB2 Advanced Copy Services (ACS)" in Data Recovery and High Availability Guide and Reference

Related reference:

"DB2 Advanced Copy Services (ACS) supported operating systems and hardware" 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:

- SQL1655C
- SQL1656C

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.

Starting in Fix Pack 3, trap resilience, is automatically applied to the load utility. In the event of a trap during a load operation, the trap will be sustained, if possible, and the table which was being loaded remains in a load pending state, just as it would if any other type of load failure occurred. Although it is possible to recover from the trap by issuing a **LOAD RESTART**, **TERMINATE**, or **REPLACE** command, the instance should still be recycled as soon as possible because the trapped thread and its descendents are left suspended. In previous releases, traps could not be sustained if the load utility was running and the load engine dispatchable units (EDUs) would crash the instance.

Related tasks:

"Recovering from sustained traps" in Troubleshooting and Tuning Database Performance

Related reference:

"db2diag - db2diag logs analysis tool " in Command Reference

FP1: Read operations on HADR standby databases are supported

Starting with Version 9.7 Fix Pack 1, you can perform read operations on your High Availability and Disaster Recovery (HADR) standby database.

Previously, use of the HADR standby database was limited to replaying the logs shipped from the primary database, and user applications could not connect to the standby database. The new functionality does not affect the primacy of log replay, so the standby continues to remain constantly ready to take over the regular database workload from the HADR primary in the case of an outage.

The main benefit of the reads on standby capability is that it improves the utilization of the HADR standby. You can run queries on the standby if they do not entail the writing of a log record. By shifting various workloads to the HADR standby, you can freeing up resources to perform more work on the primary. You can also use the standby for reporting functions.

Related concepts:

"HADR reads on standby feature" in Data Recovery and High Availability Guide and Reference

FP3: Data deduplication device support has been integrated into the backup utilities

In Version 9.7 Fix Pack 3 and later fix packs, you can use new backup parameters to specify that the target storage device supports data deduplication. These new parameters optimize the format of backup images for deduplication devices and make backup operations more efficient.

You can use the **DEDUP_DEVICE** parameter of the **BACKUP DATABASE** command or set the **iOptions** parameter of the DB2Backup API to DB2BACKUP_DEDUP_DEVICE.

In Version 9.7 Fix Pack 4 and later fix packs, this feature is also integrated in the ADMIN_CMD procedure.

Related reference:

"BACKUP DATABASE " in Command Reference

"db2Backup - Back up a database or table space" in Administrative API Reference

"BACKUP DATABASE command using the ADMIN_CMD procedure" in Administrative Routines and Views

FP2: Proxy node support for the db2adutl command has been added

Starting in Version 9.7 Fix Pack 2 and later fix packs, you can use the **OPTIONS** parameter of the **db2adutl** command to pass valid options to the Tivoli Storage Manager (TSM) server. As a result, you can use the **db2adutl** command to work in TSM environments that support client proxy nodes.

Using proxy nodes simplifies some administration tasks because you can consolidate multiple physical machines under multiple users into a single virtual node name. Proxy nodes configurations make it is easier to perform restore operations from one machine or user to another machine. During HADR scenarios, this setup also lets you access log files and restore data more easily.

Related concepts:

"Recovering data using db2adutl" in Data Recovery and High Availability Guide and Reference

Related tasks:

"Configuring a Tivoli Storage Manager client" in Data Recovery and High Availability Guide and 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.

Note: Starting with DB2 Version 9.7 Fix Pack 1, if the **diagsize** configuration parameter is set to a non-zero value and the **diagpath** configuration parameter is set to split the diagnostic data into separate directories, then the non-zero value of the **diagsize** configuration parameter specifies the total size of the combination of all rotating administration notification log files and all rotating diagnostic log files contained within a given split diagnostic data directory. For example, if a system with 4 database partitions has **diagsize** set to 1 GB and **diagpath** set to "\$n" (split diagnostic data per database partition), the maximum total size of the combined notification and diagnostic logs can reach 4 GB (4 x 1 GB).

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 Troubleshooting and Tuning Database Performance

"FP1: Diagnostic data can be stored in separate directories" on page 202

Related reference:

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

FP2: Databases can be restored using transportable sets

Starting with DB2 Version 9.7 Fix Pack 2, table spaces and SQL schemas can be restored as a set from one database to another using transportable sets.

You can also use the **db2move** command to move tables between DB2 databases.

By using the **RESTORE** command with the **TRANSPORT** option, you can restore data in a set of table spaces from a backup image into another existing database. You can

re-create database objects in the SQL schemas that reference the data in the restored table spaces. The restored table spaces and SQL schemas can function as part of the new database.

You can also use this feature to simplify the process of restoring schemas from other database solutions to DB2 Version 9.7.

Note: When you transport table spaces, a log record with a special format is created on the target database. This format cannot be read by previous DB2 versions. If you transport table spaces and then downgrade to a version earlier than DB2 Version 9.7 Fix Pack 2, then you cannot recover the target database containing the table spaces that were transported. To ensure that the target database is compatible with earlier DB2 versions, you can roll forward the target database to a point in time before the transport operation.

Related concepts:

"Database schema transporting" in Data Recovery and High Availability Guide and Reference

Related reference:

"db2move - Database movement tool " in Command Reference

FP4: New scripts improve integration between DB2 High Availability Disaster Recovery and IBM Tivoli System Automation for Multiplatforms (Windows)

In Version 9.7 Fix Pack 4 and later fix packs, you can use these new scripts to enable automatic management for DB2 servers using DB2 High Availability Disaster Recovery (HADR) and IBM Tivoli System Automation for Multiplatforms (SA MP) on Windows operating systems.

The following scripts improve the integration with Tivoli SA MP by allowing you to setup Tivoli SA MP as a cluster manager and enable Tivoli SA MP to automatically manage HADR resources:

- mkdb2
- mkhadr
- rmdb2
- hadr_start.ksh
- hadr_monitor.ksh
- hadr_stop.ksh

Restriction: You can install and run Tivoli SA MP on Windows Server 2008 systems only if they are not members of a Windows domain. Windows Server 2008 systems that are members of a Windows domain are not supported.

For implementation details of a DB2 automated failover solution using these scripts, see the "Automating DB2 HADR Failover on Windows using Tivoli System Automation for Multiplatforms" white paper at http://public.dhe.ibm.com/software/data/sw-library/db2/papers/hadr_tsa_win.pdf.

FP4: Roving high availability (HA) failover reduces downtime in partitioned database environments

Starting in Version 9.7 Fix Pack 4, you can reduce the amount of time that your data is unavailable by enabling automatic roving HA failover in partitioned database environments configured for high availability.

In N+M clustered environments with 'N' active nodes and one standby node, a failover operation occurs when one of the active nodes fails. The standby node then begins hosting the resources of the failed node. When the failed node comes back online, the clustered environment automatically turns itself offline so that the node which was originally chosen as the standby node becomes the standby node again. With roving HA failover, the last failed node in the cluster becomes the standby node without requiring additional fail back operations.

Related tasks:

"Using roving high availability (HA) failover in partitioned database environments" in Data Recovery and High Availability Guide and Reference

FP5: Automatic recovery utilities now recognize merged backups

In Version 9.7 Fix Pack 5 and later fix packs, merged backup images are fully integrated with DB2 automatic recovery utilities and database history management infrastructure.

In previous releases, automatic restore and recover did not look for merged backup images. Now, merged backups are treated as a full, non-incremental, backup by these utilities. The **RECOVER** command looks for recent database-level merged backup images, and the **RESTORE** command with the **REBUILD** option looks for recent tablespace-level merged backups. As with other types of backup images, merged backups are shown by the **LIST HISTORY** command and the **DB_HISTORY** administrative view, with an **M** indicating the operation type.

This enhancement entails a change to how backups are counted by the automatic pruning and deletion tools. Database-level merged backups are counted as full, non-incremental, database backups when DB2 determines which backups to keep and which backups to expire during a history file prune operation. As a result, if you are using merged backups, you might need to increase your current setting of the **num_db_backups** database configuration parameter, which specifies the number of database backups to retain.

FP5: db2adutl enhancements simplify backup image and log file management

In DB2 Version 9.7 Fix Pack 5 and later fix packs, you can use the **db2adutl** command to upload locally stored backup images and log files to Tivoli Storage Manager (TSM).

In previous releases and fix packs, if you wanted to save your backup images to disk and then copy them to TSM, you had to perform a series of manual steps. With the new **UPLOAD** parameter, moving these backup images to TSM is substantially easier. The **db2adutl** command also maintains TSM location information for the backup image in the recovery history file, so you can still use the **RESTORE** command with the **REBUILD WITH** parameter after uploading the image.

Other related enhancements include:

- You can use the new **SINCE** option of the **db2adutl** command to query or extract all log files greater than or equal to the specified sequence number. Using this option simplifies copying all the log files that you need to roll forward a backup to the ends of logs because you do not have to know the start and end log sequence numbers.
- You can use the **AND REMOVE** option to specify that the **db2adutl** command is to remove the archived log files or backup images after you successfully upload them
- You can use the **OLDER THAN** option to delete or upload all logs that are less than or equal to the specified sequence number.

Related reference:

"db2adutl - Managing DB2 objects within TSM " in Command Reference

FP5: ADMIN_MOVE_TABLE stored procedure now supports recoverable load

Starting in Version 9.7 Fix Pack 5, you can now specify that online table moves use recoverable load. In previous releases, you had to take a table space backup during the table move operation if you used the **COPY_USE_LOAD** option in order for the load to be recoverable.

Furthermore, this enhancement provides a significant benefit if you want to use the **ADMIN_MOVE_TABLE** procedure in a high availability disaster recovery (HADR) environment. In a standard online table move operation, the **ADMIN_MOVE_TABLE** procedure uses inserts for the **COPY** phase. This method generates one log record per row that needs to be copied, and takes regular commits. In turn, that process can slow down the operation greatly in an HADR environment because in some configurations, the primary database has to keep waiting for the standby database to send acknowledgment messages. Using the **COPY_USE_LOAD** option generates only a few log records during the **COPY** phase, greatly reducing the number of logs that are shipped to the standby system and thus avoiding a bottleneck.

Related reference:

"ADMIN_MOVE_TABLE procedure - Move tables online" in Administrative Routines and Views

FP5: Super asynchronous HADR synchronization mode has been added

The HADR synchronization mode determines when the primary database server considers a transaction complete based on the state of the logging on the standby database. Starting in Version 9.7 Fix Pack 5, you can specify **SUPERASYNC** (super asynchronous) as a synchronization mode in DB2 High Availability Disaster Recovery (HADR) environments.

The new super asynchronous mode complements the existing set of synchronization modes by ensuring that transactions can never be blocked or experience elongated response times due to network interruptions or congestion, therefore allowing transactions to be processed more quickly than in any other HADR synchronization mode.

You enable the super asynchronous mode by setting the **hadr_syncmode** database configuration parameter to SUPERASYNC.

Related concepts:

"Configuring DB2 High Availability Disaster Recovery (HADR) synchronization mode" in Data Recovery and High Availability Guide and 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” on page 68)
- Statement concentrator support (see “Statement concentrator enables access plan sharing” on page 68)
- RUNSTATS sampling improvements for statistical views (see “Statistics collection sampling performance for statistical views has been improved” on page 69)
- The ALTER PACKAGE statement for applying optimization profiles (see “Optimization guidelines for packages can be applied more easily” on page 69)
- Cost model improvements for queries in partitioned database environments (see “Cost model has been improved for queries in partitioned database environments” on page 70)

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 70)
- Scan sharing (see “Scan sharing improves concurrency and performance” on page 71)
- Partitioned indexes on partitioned tables (see “Partitioned indexes on partitioned tables improve performance” on page 27)
- Materialized query table (MQT) matching enhancements (see “Materialized query table (MQT) matching includes more scenarios” on page 74)
- The ability to store some inline LOB files in tables (see “Small LOBs can be stored in table rows and compressed” on page 30)
- 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 76)
- 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 76)
- Availability of partitioned table data during roll-out operations (see “FP1: Partitioned table data remains available during roll-out operations” on page 77)
- Queries with spatial data can run faster in partitioned database environments (see “FP2: Materialized query tables with spatial columns can be replicated” on page 78)

Another enhancement extends the support of optimization guidelines to XML data. For more information, see “Optimization profiles support guidelines for XML data” on page 32.

You can now use the **fc_parallelism** configuration parameter to control the degree of parallelism that is used for communication between members within a DB2 instance. For more information, see FCM parallelism support added ("FCM parallelism support added" on page 78).

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 Introducing DB2 Version 9.7 Fix Pack 1 Closed Beta features

"BIND " in Command Reference

"PRECOMPILE " in Command Reference

"REBIND " in Command Reference

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

"REBIND_ROUTINE_PACKAGE procedure - rebind a package" in Administrative Routines and Views

"ALTER PACKAGE " in SQL Reference, Volume 2

"ALTER_ROUTINE_PACKAGE procedure" in Administrative Routines and Views

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 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 132

"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

"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

"ALTER_ROUTINE_PACKAGE procedure" in Administrative Routines and Views

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 127

"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

"SET_ROUTINE_OPTS " in Administrative Routines and Views

"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 data 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 re-create 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 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.

Starting with DB2 V9.7 Fix Pack 1, when creating a table that uses both multidimensional clustering (MDC) and data partitioning, the system-created MDC block indexes are created as partitioned indexes. Data partitioned MDC tables can take advantage of the features available with partitioned tables such as the rolling in and rolling out of table data. For MDC tables that use table partitioning created with DB2 V9.7 and earlier, the block indexes are nonpartitioned.

Partitioned indexes over XML data

On partitioned tables, indexes over XML data that you create with DB2 V9.7 or earlier are nonpartitioned. Starting in DB2 Version 9.7 Fix Pack 1, you can create an index over XML data on a partitioned table as either partitioned or nonpartitioned. The default is a partitioned index.

To create a nonpartitioned index, specify the NOT PARTITIONED option for the CREATE INDEX statement. To convert a nonpartitioned index over XML data to a partitioned index:

1. Drop the nonpartitioned index.
2. Create index by using the CREATE INDEX statement without the NOT PARTITIONED option.

Related concepts:

"Partitioned tables" in Partitioning and Clustering Guide

"Optimization strategies for partitioned tables" in Partitioning and Clustering Guide

"Table partitioning and multidimensional clustering tables" in Partitioning and Clustering Guide

"Block indexes for MDC tables" in Partitioning and Clustering Guide

Related tasks:

"Converting existing indexes to partitioned indexes" in Partitioning and Clustering Guide

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.

Related concepts:

"Plan optimization guidelines" in Troubleshooting and Tuning Database Performance

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 127

"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

"ADMIN_IS_INLINED function - Determine if data is inlined" in Administrative Routines and Views

"ADMIN_EST_INLINE_LENGTH function - Estimate length required to inline data" in Administrative Routines and Views

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 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.

FP1: Partitioned table data remains available during roll-out operations

In DB2 Version 9.7 Fix Pack 1 and later fix packs, when detaching a data partition of a partitioned table, queries can continue to access the unaffected data partitions of the table during a roll-out operation initiated by the ALTER TABLE...DETACH PARTITION statement.

When detaching a data partition from a partitioned table using the ALTER TABLE statement with the DETACH PARTITION clause, the source partitioned table remains online, and queries running against the table continue to run. The data partition being detached is converted into a stand-alone table in the following two-phase process:

1. The ALTER TABLE...DETACH PARTITION operation logically detaches the data partition from the partitioned table.
2. An asynchronous partition detach task converts the logically detached partition into a stand-alone table.

If there are any dependent tables that need to be incrementally maintained with respect to the detached data partition (these dependent tables are referred to as detached dependent tables), the asynchronous partition detach task starts only after the SET INTEGRITY statement is run on all detached dependent tables.

In absence of detached dependents, the asynchronous partition detach task starts after the transaction issuing the ALTER TABLE...DETACH PARTITION statement commits.

The ALTER TABLE...DETACH PARTITION operation performs in the following manner:

- The DETACH operation does not wait for dynamic uncommitted read (UR) isolation level queries before it proceeds, nor does it interrupt any currently running dynamic UR queries. This behavior occurs even when the UR query is accessing the partition being detached.
- If dynamic non-UR queries (read or write queries) have not locked the partition to be detached, the DETACH operation can complete while dynamic non-UR queries are running against the table.
- If dynamic non-UR queries have locked the partition to be detached, the DETACH operation waits for the lock to be released.
- Hard invalidation must occur on all static packages that are dependent on the table before the DETACH operation can proceed.
- The following restrictions that apply to data definition language (DDL) statements also apply to a DETACH operation because DETACH requires catalogs to be updated:
 - New queries cannot be compiled against the table.
 - A bind or rebind cannot be performed on queries that run against the table.

To minimize the impact of these restrictions, issue a COMMIT immediately after a DETACH operation.

Related concepts:

"Asynchronous partition detach for data partitioned tables" in Partitioning and Clustering Guide

"Data partition detach phases" in Partitioning and Clustering Guide

Related tasks:

"Detaching data partitions" in Partitioning and Clustering Guide

Related reference:

"ALTER TABLE " in SQL Reference, Volume 2

FP2: Materialized query tables with spatial columns can be replicated

Starting in Version 9.7 Fix Pack 2, you can create replicated materialized query tables (MQTs) that contain spatial columns. Using spatial columns with replicated MQTs improves the performance of spatial queries in which spatial columns must be joined. This kind of spatial query is useful in many kinds of applications, such as risk analysis applications for the insurance industry.

The insurance industry uses risk assessment applications to determine which customers are in an area of high fire risk or high flood risk based on the customer location and a table of polygons representing fire risk or flood risk. But the customer data is often spread across multiple partitions in a partitioned database environment, which can affect the performance of these applications.

This enhancement allows the creation of a replicated MQT if the table definition includes a spatial column. Previously, this was not allowed for all user-defined structured types, of which spatial data types are a specific implementation.

FP2: Additional optimization of performance using IBM InfoSphere Optim Performance Manager Extended Edition

IBM InfoSphere® Optim™ Performance Manager Extended Edition Version 4.1 is a follow on to DB2 Performance Expert.

Starting with Version 9.7 Fix Pack 2 and later fix packs, IBM InfoSphere Optim Performance Manager Extended Edition helps optimize the performance and availability of mission critical databases and applications. IBM InfoSphere Optim Performance Manager Extended Edition delivers a proactive, comprehensive performance management approach that allows IT staff to identify, diagnose, solve and prevent performance problems in DB2 products and in associated applications including Java and DB2 Call Level Interface (CLI) applications.

FCM parallelism support added

Fast communications manager (FCM) parallelism addresses potential scalability limitations in partitioned database environments.

Such limitations can occur when workloads drive the FCM engine component to capacity, resulting in queueing delays behind internode communications processing. You can enable this feature by using the **fc_parallelism** database manager configuration parameter, which specifies the degree of parallelism that is used for communication (both control messages and data flow) between members within a DB2 instance.

Related reference:

"fcm_parallelism - Internode communication parallelism " in Database Administration Concepts and Configuration Reference

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” on page 82)
- System-defined modules, which are compatible with Oracle built-in packages (see “System-defined modules simplify SQL PL and application logic” on page 124)

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 82)
- Data type support for NUMBER, VARCHAR2, and DATE with the time component (see “Data type support has been extended” on page 83)
- The ROWNUM and ROWID pseudocolumns (see “Alternative SQL language syntax is supported” on page 84)

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 84.

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.

DB2 V9.7 Fix Pack 1 expands on the functionality of existing CLPPlus features and adds support for: bind variables, **COMPUTE** functions, and commands to help customize report presentation.

Related concepts:

"Command line processor plus (CLPPlus)" in Installing IBM Data Server Clients

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.

In Version 9.7 Fix Pack 1 and later fix packs, FORALL syntax and BULK COLLECT INTO syntax are supported in the DB2 PL/SQL compiler:

- The FORALL statement supports the INSERT, searched DELETE, searched UPDATE, and EXECUTE IMMEDIATE statements. Logically, a FORALL operation iterates over an array, processing the data change for each array element as a block operation.
- The BULK COLLECT extension to the INTO clause (in the SELECT INTO, FETCH, and EXECUTE IMMEDIATE statements) uses arrays to store multiple rows of data (either scalar types or records).

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, NVARCHAR2, and VARCHAR2 data types. You can also have the database manager interpret the DATE data type (normally composed 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.

Starting with Version 9.7 Fix Pack 2, national character support has been extended by means of the NCHAR, NVARCHAR, and NCLOB spelling for the graphic data types.

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 127

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 and NVARCHAR2 data types" in SQL Procedural Languages: Application Enablement and Support

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

"National character strings" in SQL Reference, Volume 1

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

FP4: CREATE TRIGGER statement enhancements

In Version 9.7 Fix Pack 4 and later fix packs, the CREATE TRIGGER statement has changed allowing more flexibility and functionality when creating triggers.

When creating a trigger with the CREATE TRIGGER statement, you can:

- Include more than one operation in the trigger event clause. You now have the ability to use UPDATE, DELETE, and INSERT operations together in a single clause. This capability means that the trigger is activated by the occurrence of any of the specified events. One, two, or all three trigger events can be arbitrarily specified in a CREATE TRIGGER statement. However, an operation cannot be specified more than once.
- Identify the event that activated a trigger. The trigger event predicates of UPDATING, INSERTING, and DELETING can be used as Boolean conditions for identifying trigger actions. Trigger event predicates can only be used in the trigger action of a CREATE TRIGGER statement that uses a compound SQL (compiled) statement.

Related concepts:

"Types of triggers (PL/SQL)" in SQL Procedural Languages: Application Enablement and Support

"Trigger event predicates (PL/SQL)" in SQL Procedural Languages: Application Enablement and Support

Related reference:

"CREATE TRIGGER " in SQL Reference, Volume 2

"CREATE TRIGGER statement (PL/SQL)" in SQL Procedural Languages: Application Enablement and Support

FP5: SQL compatibility has been enhanced

In Version 9.7 Fix Pack 5 and later fix packs, enhancements reduce the time and complexity of enabling some applications that you wrote for relational database products other than the DB2 product to run in a DB2 environment.

SQL compatibility has been enhanced as follows:

- Nested array and row types are supported. A nested type is a complex data type that references another complex data type. You can now nest the following types:
 - An array type as an element of an array type
 - An array or row type as a field of a row type

The maximum nesting level for arrays and row types is 16. Performing inserts or updates in deeply nested array and row types requires careful use of the syntax.

In partitioned environments, support is only available for top level SET and CALL statements that reference objects defined in nested types. Objects with nested types should not be referenced in a subquery.

- New functions are available:
 - The HEXTORAW scalar function returns a bit string representation of a hexadecimal character string.
 - The NVL2 scalar function returns a second or third argument that is based on whether the first argument is NULL or not NULL.
 - The SUBSTR2 scalar function returns a substring of a string. The SUBSTR2 function differs from the SUBSTR function in that you express the start and length arguments for the SUBSTR2 function in 16-bit UTF-16 string units (CODEUNITS16).
- Functions have been expanded:
 - The *replace-string* argument is now optional for the REPLACE scalar function.
 - In the UNNEST table function, TABLE can be used as a synonym for UNNEST. Scalar functions with an array result type can now be used as input arguments to UNNEST.
 - The ARRAY_AGG aggregate function now accepts multiple expressions.
 - The *format-string* argument is now optional for the VARCHAR_BIT_FORMAT scalar function.

Related reference:

"REPLACE " in SQL Reference, Volume 1
 "subselect" in SQL Reference, Volume 1
 "Rules for result data types" in SQL Reference, Volume 1
 "VARCHAR_BIT_FORMAT " in SQL Reference, Volume 1
 "ARRAY_AGG " in SQL Reference, Volume 1
 "CREATE TYPE (array) " in SQL Reference, Volume 2
 "VARCHAR2 and NVARCHAR2 data types" in SQL Procedural Languages: Application Enablement and Support
 "CREATE TYPE (row) " in SQL Reference, Volume 2
 "Determining data types of untyped expressions" in SQL Reference, Volume 1
 "UNNEST " in SQL Reference, Volume 1
 "HEXTORAW scalar function" in SQL Reference, Volume 1
 "NVL2 scalar function" in SQL Reference, Volume 1
 "SUBSTR2 scalar function" in SQL Reference, Volume 1

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” on page 88)
- Wild card and IP addresses support (see “Workload connection attributes have been enhanced” on page 88)
- 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 89)

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

- Work action sets can be defined at workload level (see “FP1: Work action sets can be defined at workload level” on page 94)
- Buffer pool I/O priority control for service classes (see “Buffer pool I/O priority can be controlled for service classes” on page 90)
- Integration with Linux workload management (see “DB2 workload manager supports integration with Linux workload management (WLM)” on page 90)

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 91)
- Time threshold limits unit of work duration (see “FP1: New time threshold limits unit of work duration” on page 94)
- Shorter check interval support for a subset of time-based thresholds (see “Time-based thresholds support finer granularity” on page 91)

To facilitate the migration from the deprecated DB2 Query Patroller to DB2 workload manager, you can use a script included with DB2 Version 9.7 Fix Pack 1. For more information, see Script facilitates migration from Query Patroller to workload manager.

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 92.

Additional features added in Version 9.7 improve workload monitoring. For more information, see Chapter 5, “Monitoring enhancements,” on page 37.

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 88

Chapter 5, "Monitoring enhancements," on page 37

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 on 64-bit systems, 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" on page 92

Time-based thresholds support finer granularity

The granularity for time-based thresholds has improved. This change helps reduce delays when early detection of excessive time consumption is important.

Table 6 summarizes the improvements in granularity of certain time-based thresholds in Version 9.7.

Table 6. Improved granularity of certain time-based thresholds

Threshold	Description	Granularity
CONNECTIONIDLETIME	Controls the maximum amount of time that a connection can be idle; that is, not working on a user request.	Changed from 5 minutes to 1 minute (as of Version 9.7)
ACTIVITYTOTALTIME	Controls the maximum lifetime of an activity.	Changed from 5 minutes to 1 minute (as of Version 9.7); and from 1 minute to 10 seconds (as of Version 9.7 Fix Pack 5)
UOWTOTALTIME	Controls the maximum amount of time that a unit of work can spend in the DB2 engine.	Changed from 1 minute to 10 seconds (as of Version 9.7 Fix Pack 6)

Related concepts:

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

Related reference:

"CREATE THRESHOLD " in SQL Reference, Volume 2

"ALTER THRESHOLD " in SQL Reference, Volume 2

"ALTER WORK ACTION SET " in SQL Reference, Volume 2

"CREATE WORK ACTION SET " in SQL Reference, Volume 2

"CONNECTIONIDLETIME threshold" in Workload Manager Guide and Reference

"ACTIVITYTOTALTIME threshold" in Workload Manager Guide and Reference

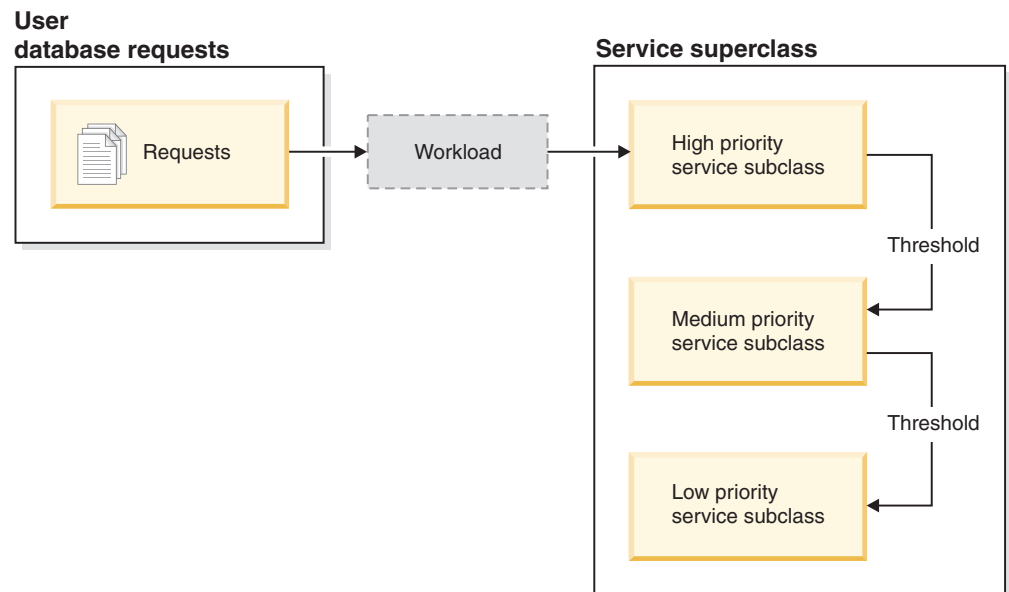
"UOWTOTALTIME threshold" in Workload Manager Guide and Reference

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 127

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

FP1: Work action sets can be defined at workload level

Starting with Version 9.7 Fix Pack 1, you can define work action sets at the workload level to control activities submitted by occurrences of those workloads, based on the type and size of work. The option to control work at the workload level complements the control options already available at the service superclass and database levels. A work action set, defined on a workload definition, applies to all work submitted by connections that are currently mapped to that workload definition.

With the ability to define work action sets at the workload level, you can monitor and control activities submitted by a specific application without having to map those activities to a distinct service class. Control of incoming work includes the application of activity thresholds to activities submitted by occurrences of the workload, as well as the ability to introduce a concurrency threshold on some or all of that same work.

The following list contains the types of work actions that are available when a work action set is applied at the workload level:

- COUNT ACTIVITY
- PREVENT EXECUTION
- COLLECT ACTIVITY DATA
- COLLECT AGGREGATE ACTIVITY DATA
- Thresholds that apply to each individual activity in the matching work class:
 - ESTIMATEDSQLCOST
 - SQLROWSRETURNED
 - ACTIVITYTOTALTIME
 - SQLTEMPSPACE
 - SQLROWSREAD
 - CPUTIME
- CONCURRENTDBCOORDACTIVITIES threshold that applies to all activities as a group in the matching work class. This threshold controls the number of concurrent activities in the matching work class from all occurrences of the workload.

Related concepts:

"Work actions and the work action set domain" in Workload Manager Guide and Reference

"Concurrency control at the workload level using work action sets" in Workload Manager Guide and Reference

FP1: New time threshold limits unit of work duration

Starting with Version 9.7 Fix Pack 1, you can use the UOWTOTALTIME threshold to specify the maximum amount of time that can elapse from the time that a unit of work first becomes active. Previously, you had to use the DB2 governor to restrict a unit of work to a specific duration.

Occasionally, an application might start transactions that run longer than a required amount of time, resulting in locks being held which prevent other more important applications from proceeding. The UOWTOTALTIME threshold triggers the termination of the long-running application or rolls back the transaction in favor of continuing progress with other work.

You can define this new threshold at the workload, service superclass, and database domains of workload management.

Related concepts:

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

"Unit of work thresholds" in Workload Manager Guide and Reference

FP1: Script facilitates migration from Query Patroller to workload manager

Starting with Version 9.7 Fix Pack 1, a sample script (qpwlmmig.pl) has been provided to facilitate the migration from the deprecated DB2 Query Patroller (QP) environment to the DB2 workload manager (WLM) environment. This script generates a file which contains the DDL statements to create the WLM objects that most closely reflect your current QP setup.

For the most part, you can continue to use the same system control approach that QP currently uses until you determine how best to use the WLM capabilities.

Related concepts:

"Query Patroller migration tool" in Workload Manager Guide and Reference

Related tasks:

"Migrating from Query Patroller to DB2 workload manager" in Upgrading to DB2 Version 9.7

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” on page 98.

The following enhancements also extend security functionality:

- AES encryption (see “AES encryption of user ID and password enhances security” on page 100)
- Transport Layer Security (see “SSL client support expanded and configuration simplified” on page 101)
- Additional Secure Sockets Layer (SSL) client support (see “SSL client support expanded and configuration simplified” on page 101)
- Ability to replay past database activities (see “FP2: Audit improvements allow replay of past database activities” on page 105)

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 101)
- Enhanced transparent LDAP authentication and group lookup support, including Kerberos authentication support on AIX operating systems (see “FP1: Transparent LDAP authentication and group lookup is supported (Linux and UNIX)” on page 103)
- Password-setting enhancements (see “Passwords can be the maximum length supported by the operating system” on page 103)
- 32-bit GSKit libraries are automatically installed (see “FP1: 32-bit GSKit libraries are included in the 64-bit DB2 product installation” on page 105)

You can control the types of operations that fenced routines can perform. For more information, see “Fenced mode process (db2fmp) privileges can be customized (Windows)” on page 104.

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. Starting in Fix Pack 5, you can further refine the privileges granted by SQLADM by setting the **DB2AUTH** registry variable to `SQLADM_NO_RUNSTATS_REORG`, which prevent users with that authority from issuing runstats and reorg operations. 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 ACCESSCTRL
- WITH DATAACCESS
- WITHOUT DATAACCESS

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 243

"Database administrator (DBADM) authority scope has changed" on page 246

"Security administrator (SECADM) abilities have been extended" on page 245

Related reference:

"DB2 server behavior changes" in Upgrading to DB2 Version 9.7

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_svcname** 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:
 - **SSLClientKeystoredb** - Set **SSLClientKeystoredb** to the fully-qualified key database file name.
 - **SSLClientKeystash** - Set **SSLClientKeystash** 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 **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.

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 **SSLClientKeystoredb** and **SSLClientKeystash**, or the client-side database manager configuration parameters **ssl_cInt_keydb** and **ssl_cInt_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_cInt_keydb**, and **ssl_cInt_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 247

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

FP1: Transparent LDAP authentication and group lookup is supported (Linux and UNIX)

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. Starting with DB2 V9.7 Fix Pack 1, transparent LDAP support has now been extended to the Linux, HP-UX, and Solaris operating systems.

Transparent LDAP authentication and group lookup allows you to use 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 operating system. The 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.

As of DB2 V9.7, transparent LDAP authentication, Kerberos authentication and group lookup are supported on AIX.

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

Fenced mode process (db2fmp) privileges can be customized (Windows)

When extended security is enabled on Windows platforms, you can change the privileges associated with the db2fmp process by using the **DB2_LIMIT_FENCED_GROUP** registry variable. You can use this feature to control the types of operations that fenced routines (such as stored procedures and UDFs) running under the db2fmp process can perform.

When you set the registry variable and add the account name associated with the DB2 service to the DB2USERS group, the operating system privileges of the db2fmp process will now include the privileges of the DB2USERS group. You can further customize the privileges of the db2fmp process by associating the DB2 service account name to other groups with special privileges.

This feature is not available if LocalSystem is used as the DB2 service account name.

Related reference:

"Miscellaneous variables" in Database Administration Concepts and Configuration Reference

FP1: 32-bit GSKit libraries are included in the 64-bit DB2 product installation

DB2 Version 9.7 Fix Pack 1 and later now installs IBM Global Security Kit (GSKit) GSKit libraries with the 64-bit DB2 database server. The 32-bit version of the GSKit libraries are automatically included when you install the 64-bit version of the DB2 database server. The GSKit libraries are necessary to support SSL connections between DB2 clients and databases.

Ensure that the path to the GSKit libraries appears in the PATH environment variable on Windows and in the LIBPATH, SHLIB_PATH or LD_LIBRARY_PATH environment variables on Linux and UNIX operating systems. For example, on Windows, add the GSKit bin and lib directories to the PATH environment variable:

```
set PATH="C:\Program Files\ibm\gsk8\bin";%PATH%  
set PATH="C:\Program Files\ibm\gsk8\lib";%PATH%
```

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

FP2: Audit improvements allow replay of past database activities

DB2 V9.7 Fix Pack 2 adds auditing capability that gives security administrators the ability to replay past database activities.

As part of a comprehensive security policy, a company might require that they retain the ability to retroactively go back a set number of years and analyze the effects of any particular request against certain tables in their database. To do this, they can institute a policy of archiving their weekly backups and associated log files such that they can reconstitute the database for any chosen moment in time. The database audit now captures sufficient information about every request made against the database to allow for the replay and analysis of any request against the relevant, restored database. This requirement covers both static and dynamic SQL statements.

Related tasks:

"Enabling replay of past activities" in Database Security Guide

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 110).
- 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 110).
- Transactions can include an increased limit of ALTER TABLE operations (see “Transactions can include an increased limit of ALTER TABLE operations” on page 111).
- 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 111).
- 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 112).
- 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 114).
- The SELECT INTO statement supports the FOR UPDATE clause (see “The SELECT INTO statement supports the FOR UPDATE clause” on page 115).

SQL programming has been enhanced with this additional functionality:

- An in-database approach to analytics, which uses a SAS embedded process on the database server (see “In-database analytics with SAS embedded process support added” on page 109)
- The TRUNCATE statement, which can quickly delete all rows from a database table (see “TRUNCATE added as an SQL statement” on page 115)
- Created temporary tables, which are a new type of user-defined temporary table (see “Created temporary tables are supported” on page 116)
- Numerous scalar functions additions and updates (see “Scalar function support has been extended” on page 116)
- 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 119)
- Choices for setting the precision of the fractional seconds of the TIMESTAMP data type (see “TIMESTAMP data type allows for parameterized precision” on page 120)
- LOB-type columns in temporary tables support (see “Temporary tables support LOB-type columns” on page 120)
- Public aliases (public synonyms) support (see “Public aliases simplify global object referencing” on page 121)
- For modules and sequences, private aliases within a schema support (see “Public aliases simplify global object referencing” on page 121)

- Global variable assignments in nested contexts support (see “FP1: Global variable assignments in nested contexts are supported” on page 173)

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

- Default values for parameters (see “Default values and named arguments promote flexibility for creating and calling procedures and functions” on page 121)
- Argument assignment by parameter name when you call procedures (see “Default values and named arguments promote flexibility for creating and calling procedures and functions” on page 121)
- Autonomous transactions, which commit work even if the invoking transaction itself is rolled back (see “Autonomous transactions are supported” on page 122)
- Application environments can be customized during the connection process by using a stored procedure (see “FP3: Application environments can be customized during the connection process” on page 123)

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 123)
- System-defined modules (see “System-defined modules simplify SQL PL and application logic” on page 124)
- An enhanced version of IBM Database Add-Ins for Visual Studio (see “IBM Database Add-Ins for Visual Studio have been enhanced” on page 126)
- 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 125)
- New DB2 sample programs as templates to create your own application programs (see “New DB2 sample programs have been added” on page 127)
- OUT and INOUT parameters support in user-defined functions (see “FP1: User-defined functions support OUT and INOUT parameters” on page 130)

SQL Procedural Language (SQL PL) programming has also been improved, see Chapter 12, “SQL Procedural Language (SQL PL) enhancements,” on page 169.

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 132).
- IBM Data Server Driver Package has been enhanced (see “ IBM Data Server Driver Package has been enhanced” on page 148).
- 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 149).
- 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 149).
- 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 150).

- The packaging of the data server drivers has been simplified (see “Component names have changed” on page 3).

In-database analytics with SAS embedded process support added

You can use an in-database approach to analytics in the data warehouse by running the SAS embedded process (SAS EP) on the DB2 database server.

The ability to dynamically score, or to run regression, clustering, neural net, and other analytic algorithms within the database server environment—thereby pushing the processing and deep analytics to where the data lives—reduces the latencies that are associated with moving data, and increases flexibility in end user analytic capabilities.

SAS Scoring Accelerator for DB2 enables the scoring process to be done inside the database and does not require data movement. Prior to Version 4.1, SAS Scoring Accelerator for DB2 translated the models that were developed by SAS Enterprise Miner into scoring functions that could be deployed inside the DB2 data server. The scoring functions could be used in SQL statements like other DB2 user-defined scalar functions, providing the following benefits:

- Reduced data movement and storage requirements
- Better data governance (most of the data stays inside the database)
- Greater computational power from the relational database management system (RDBMS)
- Better productivity through a shorter cycle of ideas to product

New enhancements in SAS Scoring Accelerator for DB2 Version 4.1, which eliminate the need to register scalar user-defined functions (UDFs), make the process of deploying and running scoring models in DB2 more dynamic, and result in better performance when executing models against large data sets. Scoring models that were developed for SAS Enterprise Miner can be deployed in a DB2 database in their native form. The models can be referenced and used in SQL statements through an analytic expression.

- The `ANALYZE_TABLE` expression, which you can specify on the table-reference clause of a subselect, enables you to efficiently execute scoring models. For more information, see “table-reference clause” or “Analyze table expressions” in the “subselect” topic.
- The `DB2_SAS_SETTINGS` registry variable enables the SAS EP. Use the `db2set` command to configure its settings. For more information, see the “Miscellaneous variables” topic.
- The SAS EP library is loaded and run in a fenced-mode process named `db2sasep`. In a partitioned database environment, this process runs on each database partition of the DB2 instance. For more information, see the “`db2ida - Stop or restart the SAS embedded process command`” topic.
- `TBFUNC` is a new explain operator type.
- When SAS queries are processing, DB2 agents might have to wait during communications with the SAS EP. This can occur when sending data to the SAS EP or when receiving data from the SAS EP. Two new in-database analytics wait times in the time spent hierarchy (`ida_send_wait_time` and `ida_rcv_wait_time`) provide visibility into the impact of these waits on overall system and query performance.

Related concepts:

"Hierarchy of time-spent monitor elements" in Database Monitoring Guide and Reference

Related reference:

"subselect" in SQL Reference, Volume 1

"db2set - DB2 profile registry " in Command Reference

"Miscellaneous variables" in Database Administration Concepts and Configuration Reference

"db2ida - Stop or restart the SAS embedded process " in

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 VIEW 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

Transactions can include an increased limit of ALTER TABLE operations

Beginning with DB2 Version 9.7, you can perform an unlimited number of ALTER TABLE statements within a single transaction. Such transactions can place the table in a reorg pending state; you can perform up to three of these types of transactions before a table reorganization is needed.

Certain ALTER TABLE operations, like dropping a column, altering a column type, or altering the nullability property of a column might put the table into a reorg pending state. In previous versions, you could execute up to three such ALTER TABLE statements before reorganization was necessary.

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 DEFERRED_FORCE.

The errors that are tolerated during object creation are limited to the following 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 about 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_INVALID=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 127

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.

NCHAR

Returns a fixed-length national character string representation of a value. This scalar function is available in Version 9.7 Fix Pack 2 and later fix packs.

NCLOB

Returns an NCLOB representation of a national character string. This scalar function is available in Version 9.7 Fix Pack 2 and later fix packs.

NEXT_DAY

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

NVARCHAR

Returns a varying-length national character string representation of a value. This scalar function is available in Version 9.7 Fix Pack 2 and later fix packs.

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.

SUBSTRB

Returns a substring of a string. This scalar function is available in Version 9.7 Fix Pack 1 and later fix packs.

TO_CLOB (which is a synonym for the CLOB scalar function)

Converts character data to the CLOB data type.

TO_NCHAR

Returns a national character representation of an input expression that has been formatted using a character template. This scalar function is available in Version 9.7 Fix Pack 2 and later fix packs.

TO_NCLOB

Returns an NCLOB representation of a character string. This scalar function is available in Version 9.7 Fix Pack 2 and later fix packs.

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

Returns a timestamp that is the expression truncated to the unit specified by the format-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 127

"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 127

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 127

Related reference:

"CREATE ALIAS " in SQL Reference, Volume 2

Default values and named arguments promote flexibility for creating and calling procedures and functions

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.

Starting with Version 9.7 Fix Pack 2, the features of default values and named arguments has been extended to user-defined functions (UDFs).

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

- 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 or UDF with added parameters using defaults such that existing invocations of the procedure or UDF 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 arguments. 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 127

Related reference:

"CALL " in SQL Reference, Volume 2

"CREATE FUNCTION (OLE DB external table) " in SQL Reference, Volume 2

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

"CREATE FUNCTION (external scalar) " in SQL Reference, Volume 2

"CREATE FUNCTION (external table) " in SQL Reference, Volume 2

"CREATE FUNCTION (sourced or template) " 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 127

Related reference:

"CREATE PROCEDURE " in SQL Reference, Volume 2

FP3: Application environments can be customized during the connection process

Starting in DB2 Version 9.7 Fix Pack 3, a procedure can be used to customize an application environment to a database from a central point of control.

The new `CONNECT_PROC` database configuration parameter is used to input the procedure name. This database configuration parameter accepts a two part procedure name. The procedure is executed implicitly by the DB2 server at the end of the connection process in order to customize a connection.

By using the connect procedure, you can now set special registers such as `CURRENT_PATH`, `CURRENT_SCHEMA`, and `CURRENT LOCALE LC_TIME`, as well as global variables without changing the application.

Related concepts:

"Customizing an application environment using the connect procedure" in Database Administration Concepts and Configuration Reference

Related reference:

"connect_proc - Connect procedure name database configuration parameter" in Database Administration Concepts and Configuration Reference

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

As of Fix Pack 1, the following extension is available:

ibm_db_django

Provides support for the Django Framework

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 IBM 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[®], Version 11.10 (and later)

Related concepts:

"Python, SQLAlchemy and Django Framework 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 7. 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.

Table 7. Supported modules (continued)

Module	Description
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.
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 email using the Simple Mail Transfer Protocol (SMTP).

Related concepts:

"System-defined modules" in SQL Procedural Languages: Application Enablement and Support

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 8. 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 2008 or Visual Studio 2010 with an administrator account or non-administrator account with elevated privileges.
- You can install the 32-bit version of the IBM Database Add-Ins for Visual Studio packaged with the 64-bit DB2 server and clients.
- The add-ins now support IBM DB2 Version 9.7 for Linux, UNIX, and Windows. 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%\sql1lib\samples` (where `%DB2PATH%` is the directory where the DB2 product is installed, as set by the **DB2PATH** environment variable)
- On UNIX operating systems: `$HOME/sql1lib/samples` (where `$HOME` is the home directory of the instance owner, as set by the **\$HOME** environment variable)

The new sample programs are as follows.

Table 9. 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. This sample program is located at: %DB2PATH%\sql11ib\samples\java\Websphere
	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 9. New Version 9.7 sample programs (continued)

Category	Feature or enhancement	Sample description
XML	Partitioned tables	The <code>xrpart.db2</code> sample program shows how to use XML in partitioned tables and how these tables support global indexes.
	Partitioned environments	The <code>xmlpartition.db2</code> sample program shows how to use XML in partitioned databases, MDC, and partitioned tables.
	XML support for MDC tables	The <code>xmlmdc.db2</code> and <code>XmlMdc.java</code> 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 <code>XmlUdfs.java</code> , <code>xmludfs.db2</code> , <code>xmludfs.sqc</code> , and <code>xmludfs.c</code> 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 <code>xmldbafn.db2</code> sample program shows how to use the <code>ADMIN_IS_INLINED</code> function to determine whether all XML documents are inlined. The program also shows how to use the <code>ADMIN_EST_INLINE_LENGTH</code> function to get the estimated inline length for XML documents that are not inlined.
	Annotated XML schema decomposition	The <code>xmldecomposition.db2</code> , <code>XmlDecomposition.java</code> , and <code>xmldecomposition.sqc</code> sample programs show how to decompose instance documents and use annotated XML decomposition to store the data in relational tables.
	Index reorganization	The <code>xmlolic.db2</code> sample program shows how to use the <code>REORG</code> command with the <code>ALLOW WRITE ACCESS</code> 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 <code>xmlindgtt.db2</code> 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 <code>db2evmonfmt.java</code> 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 9. 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 <code>wlmtiersdefault.db2</code> 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 <code>wlmtierstimerons.db2</code> script shows how to set up a DB2 WLM manager tiered service class configuration for a database. Unlike the <code>wlmtiersdefault.db2</code> script, the <code>wlmtierstimerons.db2</code> 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 <code>wlmtiersdrop.db2</code> script shows how to drop all service classes, thresholds, workloads, work classes, and work action sets that were created by the <code>wlmtiersdefault.db2</code> and <code>wlmtierstimerons.db2</code> scripts.
	Generates a script that will help migrate a Query Patroller environment to a WLM environment	The <code>qpwlmmig.pl</code> script reads the DB2 Query Patroller tables and, based on information from those tables, generates a script that contains the DDL required to create a comparable WLM setup. It also generates a second script file (<code><filename>.DROP</code>) containing the DDL to undo the changes from the first script. The <code>qpwlmmig.pl</code> script is available starting in Version 9.7 Fix Pack 1 and later fix packs.

FP1: User-defined functions support OUT and INOUT parameters

In Version 9.7 Fix Pack 1 and later fix packs, you can define external C user-defined functions (UDFs) as well as SQL PL and PL/SQL user-defined functions with OUT and INOUT parameters.

This support enables functions to return results not only through the RETURN statement but also by setting output parameters. Functions can return status information in addition to their regular results.

You can use functions with OUT or INOUT parameters as the sole expression on the right side of an assignment statement.

Related reference:

"CREATE FUNCTION (external scalar) " in SQL Reference, Volume 2

FP3: Full support for db2dsdriver configuration file extended to IBM data server clients for CLI and open source applications

Starting in Version 9.7 Fix Pack 3, the `db2dsdriver.cfg` configuration file can be used to retrieve all database connection information and properties for CLI and open source applications that use IBM Data Server Client and IBM Data Server Runtime Client.

In previous releases, IBM Data Server Client and IBM Data Server Runtime Client used the `db2dsdriver.cfg` configuration file to retrieve only Sysplex-related settings. In Version 9.7 Fix Pack 3 and later, CLI and open source applications that

use IBM Data Server Client and IBM Data Server Runtime Client can use information from the `db2dsdriver.cfg` configuration file to connect to a supported database. Other applications that use IBM Data Server Client and IBM Data Server Runtime Client, such as .NET applications or applications that use embedded SQL, can retrieve only Sysplex-related settings from this configuration file.

FP4: New IBM Data Server Driver keyword for a finer grained timeout value

Starting with DB2 Version 9.7 Fix Pack 4, the `MemberConnectTimeout` configuration keyword enables a finer grained, more precise timeout value to be set for reroute scenarios. By using the `MemberConnectTimeout` configuration keyword, the socket open will normally be faster than opening the socket with use of `ConnectionTimeout` keyword, or with no keyword at all.

The `MemberConnectTimeout` is configuration keyword is only applicable to the IBM Data Server Driver.

FP6: Environment variables now contain the driver path (Windows)

Starting with DB2 Version 9.7 Fix Pack 6, Windows system environment variables are updated to include the location of the default client interface copy for the IBM Data Server Driver Package software.

The environment variables are updated in the following cases:

- When you set the IBM Data Server Driver Package software as the default client interface copy for a new installation or upgrade
- When you switch the IBM Data Server Driver Package software so that it is the default client interface copy

The environment variables include **PATH** and **CLASSPATH**.

The environment variables are modified when the IBM Data Server Driver Package software is no longer the default client interface copy.

FP6: New utility validates a Net Search Extender text index (Windows, AIX)

Starting with DB2 Version 9.7 Fix Pack 6, you can use the Net Search Extender index validation utility (**checknseindex** command) to check whether a text index is working correctly. Use the **deepCheck** parameter to check whether Net Search Extender mapping indexes and internal indexes are consistent.

Instead of sequentially searching through text documents, Net Search Extender creates a text index. A text index consists of significant terms that are extracted from the text documents.

Related tasks:

"Validating a text index (Windows , AIX)" in Net Search Extender Administration and User's Guide

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.

The following enhancements are available in versions of the IBM Data Server Driver for JDBC and SQLJ that are shipped with DB2 Database for Linux, UNIX, and Windows.

- Version 9.7 base enhancements
- Fix Pack 1 enhancements
- Fix Pack 2 enhancements
- Fix Pack 3 enhancements
- Fix Pack 4 enhancements
- Fix Pack 5 enhancements
- Fix Pack 6 enhancements

Driver versions for Version 9.7 base enhancements

The following 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. To retrieve timestamp values with precision greater than 9, you need to use the constructors and methods in the IBM Data Server Driver for JDBC and SQLJ-only `DBTimestamp` class.

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 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 V11.10 and later data servers. 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 Informix enhancements

For connections to Informix servers, the following enhancements are added:

- Informix database names can be longer than 18 bytes.
For connections to Informix V11.11 and later, database names can be up to 128 bytes.
- Informix ISAM error reporting is enabled.
For connections to Informix 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 Informix.
For connections to Informix 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 Informix is added.
This support requires the existence of one or more Connection Managers, a primary server, and one or more alternate servers at Informix 11.50 or later.
- Workload balancing for connections to Informix is added.
For workload balancing to Informix, JDBC and SQLJ applications connect to a Connection Manager. They set the `enableSysplexWLB` property to indicate that Informix workload balancing is used.
This support requires Informix 11.50 or later.
- Support for new Informix data types is added.
As of Informix 11.50, Informix 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 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 zeros 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.

FP1: Driver versions enhancements

In DB2 Database for Linux, UNIX, and Windows Version 9.7 Fix Pack 1, the following enhancements are available in version 3.58 or version 4.8 of the driver. Version 3.58 contains JDBC 3.0 and earlier functions, and version 4.8 contains JDBC 4.0 and earlier functions.

FP1: Diagnostic information enhancements

Diagnostic information is traced to the Java standard error output stream when an exception is thrown with an SQL error code of -805. In Java database applications, -805 often indicates that all available IBM Data Server Driver for JDBC and SQLJ packages have been used because there are too many concurrently open statements. The diagnostic information contains a list of SQL strings that contributed to the exception.

FP1: Named parameter marker enhancements

JDBC named parameter marker support is enhanced to statement strings that contain SQL/PL blocks with named parameter markers.

FP1: Metadata for modules support

Methods are added to the `DB2DatabaseMetaData` class that let you retrieve information about procedures, functions, and user-defined types that are in modules.

FP2: Driver versions enhancements

In DB2 Database for Linux, UNIX, and Windows Version 9.7 Fix Pack 2, the following enhancements are available in version 3.59 or version 4.9 of the driver. Version 3.59 contains JDBC 3.0 and earlier functions, and version 4.9 contains JDBC 4.0 and earlier functions.

FP2: Extended parameter information support

Methods and constants are added to the `DB2PreparedStatement` interface, and methods are added to the `DB2ResultSet` interface that let you assign the default value or no value to a table column or a result set row. The data server must support extended indicators before you can use these methods and constants.

FP2: Property changes

The following Connection and DataSource property support is changed:

atomicMultiRowInsert

Previously, the `atomicMultiRowInsert` property did not apply to SQLJ. Now `atomicMultiRowInsert` applies to SQLJ as well as JDBC.

fetchSize

Previously, for IBM Data Server Driver for JDBC and SQLJ type 4 connectivity, and for IBM Data Server Driver for JDBC and SQLJ type 2 connectivity to DB2 Database for Linux, UNIX, and Windows data sources, the `fetchSize` property affected only scrollable cursors. Now `fetchSize` affects all types of cursors.

queryDataSize

The maximum values for the `queryDataSize` property have changed. Those values vary, depending on the data server.

FP2: DB2ParameterMetaData enhancements

The new `DB2ParameterMetaData.getProcedureParameterName` method lets you retrieve the defined name of a parameter in an SQL CALL statement.

FP2: Additional properties support

The following Connection and DataSource properties are added:

allowNullResultSetForExecuteQuery

Specifies whether the IBM Data Server Driver for JDBC and SQLJ returns null when `Statement.executeQuery`, `PreparedStatement.executeQuery`, or `CallableStatement.executeQuery` is used to execute a CALL statement for a stored procedure that does not return any result sets.

connectionCloseWithInFlightTransaction

Specifies whether the IBM Data Server Driver for JDBC and SQLJ throws an SQLException or rolls back a transaction without throwing an SQLException when a connection is closed in the middle of the transaction.

interruptProcessingMode

Specifies the behavior of the IBM Data Server Driver for JDBC and SQLJ when an application calls the Statement.cancel method.

timestampOutputType

Specifies the type of object that the IBM Data Server Driver for JDBC and SQLJ returns from a ResultSet.getTimestamp, CallableStatement.getTimestamp, ResultSet.getObject, or CallableStatement.getObject call.

FP2: Batch support enhancements

Previously, a DisconnectException with error code -4499 was thrown for IBM Data Server Driver for JDBC and SQLJ type 4 connectivity to DB2 for z/OS if the size of an update or delete batch was greater than 32KB. This restriction no longer exists, and the exception is no longer thrown.

FP2: SQLJ bind option enhancements

SQLJ program preparation now supports the SQLERROR(CHECK) bind option.

FP2: Automatically generated keys enhancements

For connections to DB2 Database for Linux, UNIX, and Windows or DB2 for z/OS, searched UPDATE, searched DELETE, and MERGE statements can return automatically generated keys (also called auto-generated keys). For UPDATE, DELETE, or MERGE statements, an automatically generated key can be any column in the table that you are updating, regardless of whether the column is generated by the data server.

FP3: Driver versions enhancements

In DB2 Database for Linux, UNIX, and Windows Version 9.7 Fix Pack 3, the following enhancements are available in version 3.61 or version 4.11 of the driver. Version 3.61 contains JDBC 3.0 and earlier functions, and version 4.11 contains JDBC 4.0 and earlier functions.

FP3: Additional properties support

The following Connection and DataSource property is added:

stripTrailingZerosForDecimalNumbers

Specifies whether the IBM Data Server Driver for JDBC and SQLJ removes trailing zeros when it retrieves data from a DECFLOAT, DECIMAL, or NUMERIC column. stripTrailingZerosForDecimalNumbers applies to JDBC and SQLJ.

FP3: Connections to DB2 for i enhancement

The IBM Data Server Driver for JDBC and SQLJ adds support for connections to DB2 for i 7.1.

FP3: DB2PreparedStatement enhancements

Two new DB2PreparedStatement methods are added.

getEstimateCost

Returns the estimated cost of an SQL statement after the statement is dynamically prepared.

getEstimateRowCount

Returns the estimated number of rows that can be returned by an SQL statement after the statement is dynamically prepared.

FP3: Caching and logging enhancements

New configuration property `db2.jcc.outputDirectory` lets you define a location in which the IBM Data Server Driver for JDBC and SQLJ stores the following files:

jccServerListCache.bin

Contains a copy of the primary and alternate server information for automatic client reroute in a DB2 pureScale® environment. This file allows primary and alternate server information to persist across JVM instances.

jccdiag.log

Contains diagnostic information that is written by the IBM Data Server Driver for JDBC and SQLJ.

connlicj.bin

Contains information about IBM Data Server Driver for JDBC and SQLJ license verification, for direct connections to DB2 for z/OS. The IBM Data Server Driver for JDBC and SQLJ writes this file when server license verification is successfully performed for a data server.

FP3: Table UDF support enhancements

The IBM Data Server Driver for JDBC and SQLJ now supports PARAMETER STYLE DB2GENERAL for Java table UDFs.

FP3: Connections to Informix enhancements

For connections to Informix servers, the following enhancements are added:

- Trusted context support is available for Informix data servers.
Trusted connections are supported for IBM Data Server Driver for JDBC and SQLJ type 4 connectivity to Informix V11.70 and later.
- An existing method is extended to support the Informix Unified Debugger.
Method `DB2Connection.setDB2ClientDebugInfo` can be called to notify the Informix data server that stored procedures and user-defined functions that are using the connection are running in debug mode.
- System monitoring support is extended to Informix data servers.
You can collect core driver time, network I/O time, server time, and application time for connections to Informix servers.

FP3: Connections to DB2 for z/OS enhancements

For connections to DB2 for z/OS Version 10 servers, the following enhancements are added:

- Full DRDA® support for Unicode

Starting with DB2 for z/OS Version 10, DRDA command and reply message parameters are sent to and received from the data server in Unicode. This support helps decrease CPU cost and character conversion errors. The IBM Data Server Driver for JDBC and SQLJ supports this enhancement for type 4 connectivity.

- Extended parameter information support
Extended parameter information support, which was added to the IBM Data Server Driver for JDBC and SQLJ in DB2 Database for Linux, UNIX, and Windows Version 9.7 Fix Pack 2, can now be used for connections to DB2 for z/OS.
- Temporal data support
DB2 for z/OS Version 10 adds support for temporal tables. This support enables automatic maintenance of historical information as a table is updated. Applications that use the IBM Data Server Driver for JDBC and SQLJ can access temporal tables. The `ResultSetMetaData.isAutoIncrement` method returns true for table columns that are defined as ROW BEGIN, ROW END, or TRANSACTION START ID.
- Binary XML support
DB2 for z/OS Version 10 supports binary XML format (Extensible Dynamic Binary XML DB2 Client/Server Binary XML Format). The IBM Data Server Driver for JDBC and SQLJ can send XML data to the data server or retrieve XML data from the data server as binary XML data.
The Connection and DataSource property `xmlFormat` specifies the format that is used to send XML data to the data server or retrieve XML data from the data server.
- Enhanced timestamp support
DB2 for z/OS Version 10 supports these new data types:
 - `TIMESTAMP(p)`, where the precision of the timestamp value, *p*, is between 0 and 12
 - `TIMESTAMP WITH TIMEZONE`
 - `TIMESTAMP(p) WITH TIMEZONE`The IBM Data Server Driver for JDBC and SQLJ adds support for update and retrieval of values in columns with these data types in JDBC and SQLJ applications.
- Support for control of EXPLAIN
DB2 for z/OS Version 10 adds support for the CURRENT EXPLAIN MODE special register, which controls the behavior of EXPLAIN with regard to eligible dynamic SQL statements. The Connection and DataSource property `currentExplainMode`, which sets CURRENT EXPLAIN MODE, now applies to connections to DB2 for z/OS.
- Support for accessing currently committed data
DB2 for z/OS Version 10 allows a read transaction to access a committed and consistent image of rows that are incompatibly locked by write transactions. The Connection and DataSource `concurrentAccessResolution` property, which controls this support, now applies to connections to DB2 for z/OS.
- Support for enhanced XML streaming
DB2 for z/OS Version 10 can return XML data to the client without having to materialize the data. This enhancement can decrease the amount of virtual storage that is needed. The IBM Data Server Driver for JDBC and SQLJ is enhanced so that Java applications automatically take advantage of this support.
- Support for dynamic statement cache enhancements

DB2 for z/OS Version 10, some dynamic SQL statements can be shared with cached statements if the only difference between the dynamic SQL statements and the cached statements is literal constants. The Connection and DataSource statementConcentrator property, which controls whether this type of statement sharing is possible, now applies to connections to DB2 for z/OS.

FP4: Driver versions enhancements

In DB2 Database for Linux, UNIX, and Windows Version 9.7 Fix Pack 4, the following enhancements are available in version 3.62 or version 4.12 of the driver. Version 3.62 contains JDBC 3.0 and earlier functions, and version 4.12 contains JDBC 4.0 and earlier functions.

FP4: Stored procedure calls with BOOLEAN parameter support

Stored procedures that are created on DB2 Database for Linux, UNIX, and Windows can have parameters of the BOOLEAN data type. IBM Data Server Driver for JDBC and SQLJ type 4 connectivity supports IN, OUT, or INOUT parameters of the BOOLEAN type in JDBC applications.

FP4: Stored procedure calls with ROW or ARRAY OF ROW parameter support

Stored procedures that are created on DB2 Database for Linux, UNIX, and Windows can have parameters of the ROW type or as an ARRAY type, in which the array elements have the ROW type. The IBM Data Server Driver for JDBC and SQLJ supports IN, OUT, or INOUT parameters of types ROW or ARRAY of ROW in JDBC applications. Applications use the Java java.sql.Struct objects for ROW parameters, and java.sql.Array objects for ARRAY of ROW parameters.

The IBM Data Server Driver for JDBC and SQLJ also introduces the DBStruct interface and DBStruct.getMetaData method for retrieving information about java.sql.Struct objects that are used for ROW parameters.

FP4: IBM Data Server Driver for JDBC and SQLJ diagnosis and trace enhancements

The following diagnosis and trace enhancements are added:

- The DB2Jcc utility tests a connection to a data server, using IBM Data Server Driver for JDBC and SQLJ type 4 connectivity or IBM Data Server Driver for JDBC and SQLJ type 2 connectivity.
- When the tracePolling configuration property is set to enable the trace while an application is running, information about all PreparedStatement objects in the application that were prepared before the trace was enabled are written to the trace destination.

FP4: Retrieval of special values enhancements

The recommended method for retrieval of data from DECFLOAT columns is to retrieve the values into java.math.BigDecimal variables. However, you cannot use the ResultSet.getBigDecimal or ResultSet.getObject method to retrieve the value NaN, Infinity, or -Infinity from a DECFLOAT column in a JDBC program, or retrieve a DECFLOAT column value into a java.math.BigDecimal variable in an SQLJ clause of an SQLJ program.

Error code -4231 is introduced to indicate that NaN, Infinity, or -Infinity was retrieved from a DECFLOAT column using the `ResultSet.getBigDecimal` or `ResultSet.getObject` method. You can test for -4231 in your applications, and retry data retrieval with the `ResultSet.getDouble` method.

FP4: Additional properties support

The following Connection and DataSource properties are added:

queryTimeoutInterruptProcessingMode

Specifies whether the IBM Data Server Driver for JDBC and SQLJ cancels the SQL statement or closes the underlying connection when the query timeout interval for a Statement object expires.

The following global configuration properties are added:

db2.jcc.sqljToolsExitJVMOnCompletion

Specifies whether the Java programs that underlie SQLJ tools such as `db2sqljcustomize` and `db2sqljbind` issue the `System.exit` call.

FP5: Driver versions enhancements

In DB2 Database for Linux, UNIX, and Windows Version 9.7 Fix Pack 5, the following enhancements are available in version 3.63 or version 4.13 of the driver. Version 3.63 contains JDBC 3.0 or earlier functions. Version 4.13 contains JDBC 4.0 or later functions, and JDBC 3.0 or earlier functions.

FP5: JDBC 4.1 support

IBM Data Server Driver for JDBC and SQLJ version 4.13 supports the following new JDBC 4.1 methods:

Class	Method
java.sql.CallableStatement	<code>getObject(int <i>parameterIndex</i>, java.lang.Class<T> <i>type</i>)</code>
	<code>getObject(java.lang.String <i>parameterName</i>, java.lang.Class<T> <i>type</i>)</code>
java.sql.Connection	<code>abort(java.util.concurrent.Executor <i>executor</i>)</code>
	<code>setSchema((java.lang.String <i>schema</i>)</code>
	<code>setNetworkTimeout(java.util.concurrent.Executor <i>executor</i>, int <i>milliseconds</i>)</code>
	<code>getSchema()</code>
java.sql.DatabaseMetaData	<code>getNetworkTimeout()</code>
	<code>generatedKeyAlwaysReturned()</code>
java.sql.Driver	<code>getPseudoColumns (java.lang.String <i>catalog</i>, java.lang.String <i>schemaPattern</i>, java.lang.String <i>tableNamePattern</i>, java.lang.String <i>columnNamePattern</i>)</code>
	<code>getParentLogger()</code>
	<code>getPseudoColumns (java.lang.String <i>catalog</i>, java.lang.String <i>schemaPattern</i>, java.lang.String <i>tableNamePattern</i>, java.lang.String <i>columnNamePattern</i>)</code>
java.sql.Statement	<code>abort(java.util.concurrent.Executor <i>executor</i>)</code>
	<code>closeOnCompletion()</code>
	<code>isCloseOnCompletion()</code>
javax.sql.CommonDataSource	<code>getParentLogger()</code>

IBM Data Server Driver for JDBC and SQLJ version 4.13 supports the following JDBC 4.1 changes to JDBC methods:

Class	Method	Change
java.sql.DatabaseMetaData	getColumns	In JDBC 4.0 or earlier, the result set that getColumns returns contains a column named SCOPE_CATALOG. In JDBC 4.1 or later, the name of that column is SCOPE_CATALOG.

IBM Data Server Driver for JDBC and SQLJ version 4.13 supports the following JDBC 4.1 changes to data type mappings for updating table columns:

Java data type	Database data type
java.math.BigInteger	BIGINT
java.util.Date	CHAR, VARCHAR, DATE, TIME or TIMESTAMP
java.util.Calendar	CHAR, VARCHAR, DATE, TIME or TIMESTAMP

IBM Data Server Driver for JDBC and SQLJ version 4.13 supports the following JDBC 4.1 escape syntax, which you can use to limit the number of rows that are retrieve from a table:

```
{limit integer}
```

For example the escape clause in the following query tells JDBC to return at most 20 rows from the EMPLOYEE table:

```
stmt.executeQuery("SELECT EMPNO FROM EMPLOYEE {limit 20}");
```

FP5: Trace enhancements

Circular tracing is introduced for the IBM Data Server Driver for JDBC and SQLJ. Circular tracing means that there are a fixed number of trace output data sets, and that each data set has a fixed size. New trace data overwrites old trace data when all data sets are full. Circular tracing is an alternative to sequential tracing, which results in trace output files that grow indefinitely.

FP5: Statement caching enhancements

IBM Data Server Driver for JDBC and SQLJ internal statement caching can improve the performance of Java database applications. Internal statement caching is introduced for connections that use the java.sql.DriverManager or com.ibm.db2.jcc.DB2SimpleDataSource interfaces. Previously, internal statement caching was available only for connections that used the javax.sql.ConnectionPoolDataSource or javax.sql.XADataSource interfaces.

FP5: Enhancements to stored procedure calls with ROW or ARRAY OF ROW parameter support

In DB2 Version 9.7 Fix Pack 4, the IBM Data Server Driver for JDBC and SQLJ added support for IN, OUT, or INOUT parameters of types ROW or ARRAY of ROW in JDBC applications. In DB2 Version 9.7 Fix Pack 5, the IBM Data Server Driver for JDBC and SQLJ adds support for the following types of nesting:

- ARRAY parameters with ARRAY elements
- ARRAY parameters with ROW elements
- ROW parameters that contain ARRAY types
- ROW parameters that contain ROW types

FP5: Additional properties support

The following Connection and DataSource properties are added:

com.ibm.db2.jcc.DB2SimpleDataSource.maxStatements

Controls the internal statement cache that is associated with a Connection object. Setting maxStatements to a positive value for a new connection enables the internal statement cache, and specifies the maximum number of statements in the cache.

traceFileCount

Specifies the maximum number of trace files, for circular tracing.

traceFileSize

Specifies the maximum size of each trace file, for circular tracing.

traceOption

Specifies whether sequential tracing or circular tracing is done.

useJDBC41DefinitionForGetColumns

Specifies whether the IBM Data Server Driver for JDBC and SQLJ honors the JDBC 4.1 change of getColumns result set column name SCOPE_CATALOG to SCOPE_CATALOG.

The following global configuration properties are added:

db2.jcc.traceFileCount

Specifies the maximum number of trace files, for circular tracing. This property provides the default for Connection and DataSource property traceFileCount.

db2.jcc.traceFileSize

Specifies the maximum size of each trace file, for circular tracing. This property provides the default for Connection and DataSource property traceFileSize.

db2.jcc.traceOption

Specifies whether sequential tracing or circular tracing is done. This property provides the default for Connection and DataSource property traceOption.

FP5: Connections to DB2 for z/OS enhancements

For connections to DB2 for z/OS, the following enhancement is added:

- **RACF[®] password phrase support:** For greater security, Java database applications can supply a RACF password phrase instead of a simple password for password authentication or encrypted password authentication. A password phrase is a character string that consists of mixed-case letters, numbers, and special characters, including blanks. A password phrase can be between 9 and 100 characters long, or between 14 and 100 characters long, depending on the RACF setup.

FP5: Connections to DB2 for IBM i enhancements

For connections to DB2 for i 7.1 and later servers, the following enhancements are added to the IBM Data Server Driver for JDBC and SQLJ:

- Three-part name support

- XML data type support
- ARRAY type support
- SSL authentication support
- AES encryption support

FP6: Driver versions enhancements

In DB2 Database for Linux, UNIX, and Windows Version 9.7 Fix Pack 6, the following enhancements are available in version 3.64 or version 4.14 of the driver. Version 3.64 contains JDBC 3.0 or earlier functions. Version 4.14 contains JDBC 4.0 or later functions, and JDBC 3.0 or earlier functions.

FP6: Alternate group support

Alternate group support allows the IBM Data Server Driver for JDBC and SQLJ to move an application workload to a DB2 for z/OS alternative data sharing group or a DB2 Database for Linux, UNIX, and Windows alternative DB2 pureScale instance when the primary group is unavailable.

Important: You need to apply APAR IC79084 after you install DB2 9.7 fix pack 6 to make alternate group support available on your system.

You enable alternate group support by providing the addresses of alternate groups in Connection or DataSource properties.

The Connection or DataSource properties are:

- `alternateGroupServerName`
- `alternateGroupPortNumber`
- `alternateGroupDatabaseName`

In addition, you can enable or disable seamless failover behavior for alternate group support by setting the `enableAlternateGroupSeamlessACR` Connection or DataSource property.

FP6: Connections to DB2 for z/OS enhancements

For connections to DB2 for z/OS, the following enhancements are added:

- **New and changed Connection and DataSource properties:**

securityMechanism

The `CLIENT_CERTIFICATE_SECURITY` value is added to enable certificate authentication for connections to a DB2 for z/OS Version 10 data server when the data server and the IBM Data Server Driver for JDBC and SQLJ are configured for SSL authentication.

currentLocaleLcCtype

The `currentLocaleLcCtype` property is added for connections to DB2 for z/OS data servers to specify the `LC_CTYPE` locale that is used to execute SQL statements that use a built-in function that references a locale.

FP6: Additional properties support

The following Connection and DataSource properties are added:

alternateGroupPortNumber

Specifies the port numbers for alternate groups to which an application can connect.

alternateGroupServerName

Specifies the host names for alternate groups to which an application can connect.

alternateGroupDatabaseName

Specifies the database names for alternate groups to which an application can connect.

implicitRollbackOption

Specifies the actions that the IBM Data Server Driver for JDBC and SQLJ takes when a transaction encounters a deadlock or a timeout.

Related concepts:

“New DB2 sample programs have been added” on page 127

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

"The IBM_DB Ruby driver and Rails adapter" in Getting Started with Database Application Development

"Command line processor plus (CLPPlus)" 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.

Call level interface (CLI) functionality has been enhanced

Version 9.7 includes new enhancements that extend the CLI functionality that can 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 before fetching.

Restriction: This feature is not supported if the cursor contains LOBs or XML data.

Before 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 in Version 9.5 Fix Pack 3 and later fix packs.

CLI dynamic packages can be bound on demand

You can use the new `SQLCreatePkg` API to bind arbitrary packages to a database. You can control some **BIND** options by using this API.

This support is also available in Version 9.5 Fix Pack 3 and later fix packs.

CLI ping capabilities have been enhanced

CLI applications can now override the default packet size that is used to ping a database and specify the number of times to ping the database before yielding a final result.

Before 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 the `SQL_ATTR_PING_NTIMES` attribute, 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 and pass `SQL_ATTR_PING_NTIMES` as the attribute argument.

This support is also available in Version 9.5 Fix Pack 3 and later fix packs.

Anyorder file type modifier can help improve the performance of CLI applications that use the LOAD API

You can help improve the performance of CLI applications that use the LOAD API by using the new `SQL_ATTR_LOAD_MODIFIED_BY` statement attribute to specify the anyorder file type modifier. Use the statement attribute to specify multiple file type modifiers that are 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);
```

Header information in CLI traces can be suppressed

You can suppress header information that is typically displayed in a CLI trace by setting the new `SQL_ATTR_TRACENOHEADER` environment attribute 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 example is displayed in the CLI trace 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 that access DB2 for z/OS can now roll back a transaction even while in `SQL_NEED_DATA` state by setting the new `SQL_ATTR_FORCE_ROLLBACK` connection attribute by using the `SQLSetConnectAttr` API. This behavior is supported when the **StreamPutData** CLI/ODBC configuration keyword is set to 1.

Before this enhancement, CLI applications running on DB2 for z/OS had to drop and reestablish the database connection to come out of the `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 the Dynamic Data Format, CLI applications can now call for previously accessed LOB columns, and maintain the data offset position from the previous `SQLGetData()` call. You control this behavior by specifying the new **AllowInterleavedGetData** CLI/ODBC configuration keyword or the new `SQL_ATTR_ALLOW_INTERLEAVED_GETDATA` statement attribute.

Before this enhancement, CLI applications could call `SQLGetData()` for previously accessed LOB, if you specified 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 supported:

```
CALL addEmp(?,?,?,?);
CALL addEmp(:empNo, :empName, :empDeptNo, :empAddr);
```

Before this enhancement, you could not pass procedure arguments in an order that differed from the order in which you defined the parameters when creating the procedure.

CLI provides no support to bind by name. CLI processes anything that matches a valid parameter marker, and treats it as if it is a normal parameter marker, which is 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 IBM Data Server Driver for ODBC and CLI for all servers.

CLI applications support default parameter values

When you use the CALL statement to call a procedure, you no longer have to specify values for all the parameters. Unspecified parameters take the default values that you defined for the procedure.

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 you use CLI array input chaining.

The behavior of cursor stability scans in CLI applications can be controlled

You can now use the **ConcurrentAccessResolution** CLI/ODBC configuration keyword to specify a prepare attribute that overrides the behavior specified for cursor stability (CS) 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 semantic that is defined by the **cur_commit** configuration parameter.

CLI applications support additional data type conversions and variable-length TIMESTAMP data type

CLI applications now support conversions between the following data types:

Table 10. Support for additional data type conversions in 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_TYPE_TIMESTAMP
SQL_TYPE_TIMESTAMP	SQL_C_CHAR

In addition, CLI performs the conversions that are 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 `SQL_ATTR_REPORT_TIMESTAMP_TRUNC_AS_WARN` CLI statement attribute is also available to control whether a datetime overflow results in an error (SQLSTATE 22008) or warning (SQLSTATE 01S07).

FP3: APIs to add and drop databases

CLI applications can now use the `SQLCreateDb()` and `SQLDropDb()` APIs to add and drop databases. You can use the equivalent `SQLCreateDbW()` and `SQLDropDbW()` APIs with UNICODE CLI applications for adding and dropping databases.

DB2 Database servers must connect to the database instance by using the **ATTACH** configuration keyword.

FP3: Enhanced connection support

The new **ATTACH** configuration keyword allows `SQLDriverConnect()` to attach to a server instance instead of a database. CLI applications can now use this configuration keyword when connecting to a DB2 Linux, Unix, and Windows database server.

FP3: Code page conversion can be disabled during bind-in and bind-out operations

The new connection level attribute `SQL_ATTR_OVERRIDE_CHARACTER_CODEPAGE` enables CLI applications to specify a database code page even when the code page is not available at the client end. When this new attribute is set to non-zero value, CLI will skip code page conversion during bind-in or bind-out of character data. CLI will fetch/insert the raw data without performing conversion from/to the server.

FP3: Network connection statistics support

Using the new `SQL_ATTR_NETWORK_STATISTICS` connection attribute, CLI applications can collect the following network statistics for a database connection:

- Database processing time
- Total round-trip elapsed time
- Number of bytes that are sent to the database server
- Number of bytes that are received from the database server
- Number of DRDA round trips

FP3: Enhancement to DB2 for z/OS Version 10 feature support

Starting in Version 9.7 Fix Pack 3a, CLI applications support DB2 for z/OS Version 10, which includes support for the following features:

- Ability for CLI applications to use statement cache for dynamic statement containing literal values.

The `statementConcentrator` property for connection and statement attributes, which controls whether statement cache is used for dynamic statement containing literal values, now applies to connections to DB2 for z/OS Version 10.

- Ability for CLI applications to use new `TIMESTAMP_WITH_TIMEZONE` data type.

DB2 for z/OS Version 10 supports the new `TIMESTAMP_WITH_TIMEZONE` data type. The new `TIMESTAMP_WITH_TIMEZONE` data type is only available in new function mode.

- Ability for CLI applications to use the `SQL_ATTR_EXTENDED_INDICATORS` statement attribute.

The `SQL_ATTR_EXTENDED_INDICATORS` statement attribute that eliminates the need to indicate the position in the SQL statement can now be used by CLI applications for connections to DB2 for z/OS Version 10.

- The **DB2Explain** CLI configuration keyword is available for use with the DB2 for z/OS Version 10 server.

DB2Explain CLI configuration keyword support which is available for DB2 for Linux, UNIX, and Windows, data server is now extended to DB2 for z/OS Version 10 servers.

- Support for a currently committed semantic through `SQL_ATTR_CONCURRENT_ACCESS_RESOLUTION` attribute or **ConcurrentAccessResolution** CLI configuration keyword.

Currently committed semantic support which is available for DB2 Linux, UNIX, and Windows data server is now extended to DB2 for z/OS Version 10 servers. The z/OS server, however, currently only supports query against uncommitted INSERT and uncommitted DELETE.

- Support for binary XML format.

The DB2 for z/OS Version 10 New Function Mode supports the binary XML format. CLI now provides a pass through mechanism for Binary XML data format.

FP4: New CLI connection attribute, attribute value and changes

Version 9.7 Fix Pack 4 and later fix packs include the following enhancements to the CLI:

- The `SQL_ATTR_NETWORK_STATISTICS` connection attribute has a new value, `SQL_NETWORK_STATISTICS_ON_SKIP_NOSERVER`. In addition to enabling network statistics collection for a connection, this option omits network flows that are known to have no server time reported, for example `COMMIT` and `ROLLBACK` statements.
- The `SQLGetInfo` function has a new `InfoType` value, `SQL_DRIVER_BLDLEVEL`, which returns information about the build level for the current version of CLI.
- You can use the CLI asynchronous execution and load processing features together by specifying the `SQL_ATTR_ASYNC_ENABLE` and `SQL_ATTR_USE_LOAD_API` attributes at the same time.

FP4: New enhancement to automatic client reroute and archive option for the db2diag command

Version 9.7 Fix Pack 4 and later fix packs include the following enhancements to the CLI:

- As an aid to maintaining a current list of available servers for automatic client reroute, if the `db2dsdriver.cfg` file has no alternate servers defined in the `<acr>` section, at the first successful connection to the server, the client creates a local cache file, `svr1st.xml`, and updates it with the server's list of available alternate servers. This file is refreshed whenever a new connection is made and the server's list differs from the contents of the client `svr1st.xml` file.

When you modify the `db2dsdriver.cfg` file, your CLI application can call the `SQLReloadConfig` function to validate the entries for all alternate servers within the `<acr>` section. For each server, an attempt is made to open a socket by using the specified host name and port. If all servers in the alternate server list for an active database connection are unreachable, an error message is returned in the `DiagInfoString` argument of the `SQLReloadConfig` function.

- The **db2diag** command with `-archive` option is available for IBM Data Server Driver Package and IBM Data Server for ODBC and CLI. This command option enables you to archive the diagnostic log file on an instance-less client.

FP4: New features available for DB2 on Windows operating system

Version 9.7 Fix Pack 4 and later fix packs include the following enhancements to the CLI:

- The DB2 interactive CLI command (**db2cli**) has the new parameter, **install**. On Windows operating system, you can use the **-setup** and **-cleanup** options of this command parameter to register or unregister the IBM Data Server Driver for ODBC and CLI.

The **validate** option of the **db2cli** command has been enhanced so that invalid keywords found in the `db2cli.ini` and `db2dsdriver.cfg` files are displayed as well as valid keywords.

- On Windows operating systems, when you are using the IBM Data Server Driver for ODBC and CLI, the **db2diag.log** file is located in `%UNZIPPED_PATH%\IBM\DB2\`. (In previous release, the location was `%UNZIPPED_PATH%\IBM\DB2\CLIDRIVER\`.)

FP5: Support for IBM i DB2 servers has been enhanced

Starting in Version 9.7 Fix Pack 5, CLI applications support the following features in IBM i DB2 servers:

- The SQL_XML data type is supported with DB2 for i V7R1. See XML data handling in CLI applications .
- The SQL_ATTR_EXTENDED_INDICATORS connection attribute is available for connections to IBM i DB2 servers. See SQL_ATTR_EXTENDED_INDICATORS.
- Support to facilitate the migration from iAccess drivers to CLI. See SchemaList CLI/ODBC configuration keyword or SchemaFilter IBM Data Server Driver configuration keyword.

FP5: CLI optimizations to improve performance, troubleshooting, and high availability

In Version 9.7 Fix Pack and later fix packs, CLI applications support the following features to improve application performance, troubleshooting, and high availability:

- Support for implicit COMMIT after a complete result set is read from a cursor. See SQL_ATTR_COMMITONEOF.
- Support to convert array input chaining into a column-wise array insert for applications that are connected to DB2 for z/OS servers. See SQL_ATTR_COLUMNWISE_MRI.
- Support to return the number of rows in a table that are affected by each parameter set in applications that use array input to achieve bulk inserts, deletes, or updates. See SQL_ATTR_PARC_BATCH.
- Support to fetch or insert data without code page conversions. See SQL_ATTR_OVERRIDE_CODEPAGE.
- Use of a password phrase to access DB2 for z/OS servers. A password phrase is a character string that consists of mixed-case letters, numbers, and special characters including blanks. See PWD CLI/ODBC configuration keyword or Password IBM Data Server Driver configuration keyword.
- Support to collect statistics for server time reported on COMMIT and ROLLBACK. See SQL_ATTR_NETWORK_STATISTICS.
- Support for prefixing warning messages with a diagnostic string consisting of the database connection information when you call the SQLReloadConfig () function. See SQLReloadConfig function.
- Support for returning a warning message on seamless failover. See SQL_ATTR_REPORT_SEAMLESSFAILOVER_WARNING.
- Support of alternate groups for DB2 for Linux, UNIX, and Windows servers and DB2 for z/OS servers. See Alternate groups for connections to DB2 Database for Linux, UNIX, and Windows from non-Java clients.

FP5: db2cli command changes

In Version 9.7 Fix Pack 5 and later fix packs, the following changes apply to the **db2cli** command:

- The **validate** parameter now shows the copy name and type in the command output. See Validating IBM Data Server Driver Package (Windows) installation.
- Support for adding data sources from the local database directory by using the **db2cli** command with the **registerdsn -add** parameter. See the db2cli - DB2 interactive CLI command.
- Support for adding or modifying data source, database, and parameter entries in the db2dsdriver.cfg configuration file by using the **db2cli** command with the new **writecfg** parameter. See the db2cli - DB2 interactive CLI command.

FP5: ODBC 3.8 support

In Version 9.7 Fix Pack 5 and later fix packs, the following changes or additions have been made to support ODBC 3.8:

- The new `SQL_OV_ODBC3_80` value for the `SQL_ATTR_ODBC_VERSION` environment attribute. See `SQL_ATTR_ODBC_VERSION`.
- A new connection attribute for better management of connection pooling, ODBC 3.8. See `SQL_ATTR_RESET_CONNECTION`.
- A new transaction state called suspended state for better application programming and transaction control. See `SQLEndTran` function

FP6: Prefetched cursor support for seamless automatic client reroute

In Version 9.7 Fix Pack 6 and later fix packs, if all the data, including the end of file (EOF) character, is returned in the first query block or in a subsequent fetch request, the CLI driver can perform seamless failover when you issue a `COMMIT` or `ROLLBACK` statement after the server becomes unreachable. For seamless failover to take place, the following conditions must be met:

- You must enable both the `enableAcr` and `enableSeamlessAcr` parameters.
- The cursor must have blocking enabled.
- The cursor must be either read only or forward only.

See Operation of automatic client reroute for connections to DB2 Database for Linux, UNIX, and Windows from non-Java clients, Operation of automatic client reroute for connections from non-Java clients to DB2 for z/OS servers, and Operation of automatic client reroute for connections to Informix database server from non-Java clients.

FP6: New enhancements to the DB2 interactive CLI command (`db2cli`) (Windows)

Version 9.7 Fix Pack 6 and later fix packs include the following enhancements to the DB2 interactive CLI command (`db2cli`):

- You can use the `db2cli validate` command to obtain a complete list of IBM data server client packages that are installed on a Windows operating system. When you issue the command on a Windows operating system, the `db2cli` command reports all duplicate and conflicting settings for the same property in the same DSN, database, or global section of the `db2dsdriver.cfg` file.
- You can use the `db2cli validate` command to check duplicate or conflicting keyword entries in the same section of the `db2dsdriver.cfg` file.
- On Windows operating systems, new options are added to the `registerdsn` parameter for the `db2cli` command:
 - The `db2cli registerdsn -remove` command include the following new options:
 - `-alldsn`
 - `-copyname copy_name`
 - `-allcopies`
 - `-force`
 - `-dsn`
 - The `db2cli registerdsn -add` command includes the new `-dsn` option.
 - The `db2cli registerdsn -list` command includes the following new options:

- -copyname *copy_name*
- -allcopies

See **db2cli** - DB2 interactive CLI command .

FP6: Support for DB2 for i servers has been further enhanced

In Version 9.7 Fix Pack 6 and later fix packs, CLI applications support the following features in DB2 for i servers:

- SQL_BINARY and SQL_VARBINARY data types
- The following client information properties, in DB2 for i V6R1 and later:
 - SQL_ATTR_INFO_ACCTSTR
 - SQL_ATTR_INFO_APPLNAME
 - SQL_ATTR_INFO_USERID
 - SQL_ATTR_INFO_PROGRAMID
 - SQL_ATTR_INFO_WRKSTNNAME

See Environment attributes (CLI) list and Connection attributes (CLI) list.

FP6: SQL_ATTR_NETWORK_STATISTICS attribute support has been further enhanced

In Version 9.7 Fix Pack 6 and later fix packs, you can obtain the server time for COMMIT or ROLLBACK SQL operations on DB2 for z/OS Version 10 and later. See Connection attributes (CLI) list.

FP6: New LDAP keywords for the db2dsdriver.cfg file

In Version 9.7 Fix Pack 6 and later fix packs, the following new keywords are added to the db2dsdriver.cfg for the LDAP support:

- **EnableLDAP**
- **LDAPServerHost**
- **LDAPServerport**
- **ClientProvider**
- **BaseDN**
- **UserID**
- **Password**

See IBM Data Server Driver configuration keywords.

FP6: New SQL_C_CURSORHANDLE C data type

In Version 9.7 Fix Pack 6 and later fix packs, the new SQL_C_CURSORHANDLE C data type is available for use with the SQL_CURSORHANDLE SQL data type. See SQL symbolic and default data types for CLI applications.

FP6: QueryTimeout keyword support

In Version 9.7 Fix Pack 6 and later fix packs, the CLI driver now supports use of the **QueryTimeout** keyword in the db2dsdriver.cfg file. See **QueryTimeout** IBM Data Server Driver configuration keyword.

Related concepts:

"Diagnostic tools" in DB2 Connect User's Guide

"Analyzing db2diag log files using db2diag tool" in Troubleshooting and Tuning Database Performance

"The DB2 database application development environment" in Getting Started with Database Application Development

"db2oreg1.exe overview" in Call Level Interface Guide and Reference, Volume 1

"Configuration of DB2 Database for Linux, UNIX, and Windows automatic client reroute support for non-Java clients" in Call Level Interface Guide and Reference, Volume 1

"Example of enabling DB2 Database for Linux, UNIX, and Windows automatic client reroute support in non-Java clients" in Call Level Interface Guide and Reference, Volume 1

"Example of enabling DB2 Database for Linux, UNIX, and Windows workload balancing support in non-Java clients" in Call Level Interface Guide and Reference, Volume 1

Related tasks:

"Importing data with the CLI LOAD utility in CLI applications" in Call Level Interface Guide and Reference, Volume 1

"Executing functions asynchronously in CLI applications" in Call Level Interface Guide and Reference, Volume 1

"Registering the IBM Data Server Driver for ODBC and CLI with the Microsoft ODBC driver manager" in Call Level Interface Guide and Reference, Volume 1

"Registering the IBM Data Server Driver for ODBC and CLI with the Microsoft DTC" in Call Level Interface Guide and Reference, Volume 1

"Uninstalling the IBM Data Server Driver for ODBC and CLI" in Call Level Interface Guide and Reference, Volume 2

"Registering ODBC data sources for applications that use the IBM Data Server Driver for ODBC and CLI" in Call Level Interface Guide and Reference, Volume 1

Related reference:

"SQLColAttribute function (CLI) - Return a column attribute" in Call Level Interface Guide and Reference, Volume 2

"SQLGetInfo function (CLI) - Get general information" in Call Level Interface Guide and Reference, Volume 2

"db2cli - DB2 interactive CLI " in Command 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

"db2diag - db2diag logs analysis tool " in Command Reference

"Diagnostic support in the IBM Data Server Driver for ODBC and CLI" in Call Level Interface Guide and Reference, Volume 1

IBM Data Server Provider for .NET is enhanced

Version 9.7 includes enhancements that improve IBM Data Server Provider for .NET support and connectivity to other data servers.

ARRAY data type support

ARRAY data type support is added to IBM Data Server Provider for .NET. You can use the ARRAY data type with your stored procedure parameters. You can bind an array to a parameter in your procedure as a single argument. This support simplifies the code around your SQL statements.

Compound statement support

Compound statements are supported by IBM Data Server Provider for .NET. Using compound statements in your SQL statements can improve performance by having the statements use the same access plan for a group of statements.

Host variable support

Host variable support is added to IBM Data Server Provider for .NET to improve compatibility with applications that you use with other data servers. You can use host variables (:param) in place of positioned or named parameter markers (@param). However, you can specify only one type of parameter in a particular statement at a time.

Variable length TIMESTAMP support

IBM Data Server Provider for .NET now supports variable-length time stamps. This support makes it easier to work with other data servers. Previously, the TIMESTAMP data type had a fixed six-digit precision. The TIMESTAMP data type now supports 0 - 12 digits of precision.

FP1: Statement concentrator disabling support

Starting in Version 9.7 Fix Pack 1, you can use added bypass literal properties to disable statement concentration for dynamic statements. There are StatementConcentrator properties for the DB2Command and DB2ConnectionStringBuilder classes, a connection string parameter, and a db2dsdriver.cfg file keyword.

FP1: DATE and TIMESTAMP literals support

Starting in Version 9.7 Fix Pack 1, you can bind in string objects with TIMESTAMP values into DATE and TIME columns and bind in string objects with DATE values into TIMESTAMP columns.

FP2: 32-bit drivers included in 64-bit package

Starting in Version 9.7 Fix Pack 2, the 32-bit versions of IBM Data Server Provider for .NET are included in the 64-bit package. When you install the 64-bit drivers, the 32-bit drivers are also installed, in a separate directory named, sql1lib\bin\netf20_32.

FP2: Database connection synonym processing bypass support

Starting in Version 9.7 Fix Pack 2, you can use a new db2dsdriver.cfg file keyword or connection string property, SkipSynonymProcessing, to bypass synonym processing when opening a connection. Using the keyword or connection string property when you do not require synonym processing can reduce

connection time overhead when you use `DB2Connection` or `DB2ConnectionStringBuilder`.

FP2: Query timeout support in the `db2dsdriver.cfg` file

Starting in Version 9.7 Fix Pack 2, you can use a new `db2dsdriver.cfg` file keyword, `QueryTimeout`, as a centralized control to indicate how long a client should wait for a query to run before timing out.

FP2: Extended indicator support for default and unassigned parameters

Starting in Version 9.7 Fix Pack 2, you can set named and positioned parameters to use the default or unassigned values as defined by the data server.

FP2: Enhanced CALL statement support

Starting in Version 9.7 Fix Pack 2, you can use named arguments in any order within CALL statements. Named arguments can work with host variables and positioned parameters, however, named parameters are not supported.

FP2: Module support

Starting in Version 9.7 Fix Pack 2, IBM Data Server Provider for .NET includes support for modules. A module is a collection of database objects such as functions, procedures, and variables.

FP3: Supports DB2 for z/OS Version 10

Starting in Version 9.7 Fix Pack 3, IBM Data Server Provider for .NET supports DB2 for z/OS Version 10, which includes support for client disabled dynamic statement cache, extended indicators, Timestamp with timezone, and new Explain features.

FP3: `DB2Type.Cursor` support

Starting in Version 9.7 Fix Pack 3, IBM Data Server Provider for .NET introduces a new member for the `DB2Type` enumeration called `Cursor`. This member should be used when binding an output parameter of the type `cursor`

FP3: Trusted Context support

Starting in Version 9.7 Fix Pack 3, IBM Data Server Provider for .NET adds support for IBM Informix database server Version 11.70.

FP4: Updated Canonical functions

Starting in Version 9.7 Fix Pack 4, IBM Data Server Provider for .NET supports new canonical functions.

FP4: New `testconn` utility

Starting in Version 9.7 Fix Pack 4, IBM Data Server Provider for .NET supports new `testconn` utility. New `Testconn40.exe` can be used to validate the .NET provider with a .NET Framework 4.0.

FP4: Framework 4.0 support

Starting in Version 9.7 Fix Pack 4, IBM Data Server Provider for .NET supports .NET Framework 4.0.

FP4: Visual Studio 2010 support

Starting in Version 9.7 Fix Pack 4, IBM Visual Studio Add-ins support Visual Studio 2010.

FP4: FitHighPrecisionType support

Starting in Version 9.7 Fix Pack 4, IBM Data Server Provider for .NET supports a new keyword FitHighPrecisionType.

FP4: Removal of U2 support

Starting in Version 9.7 Fix Pack 4, IBM Data Server Provider for .NET and IBM Visual Studio Add-ins no longer support U2 servers.

FP5: Application development enhancements

Starting in Version 9.7 Fix Pack 5, the following features are added to facilitate application development:

- Support for implicit COMMIT after reading a complete result set from a cursor.
- Support of the new password keyword in the db2dsdriver.cfg file.
- You can use a password phrase as the password when accessing DB2 for z/OS servers. A password phrase is a character string that consists of mixed-case letters, numbers, and special characters including blanks.
- Support for adding data sources from the local database directory by using the **db2cli** command with the **registerdsn -add** parameter.
- Support for adding and modifying data sources or database entries, as well as adding parameters to the common section of the db2dsdriver.cfg file by using the **db2cli** command with the **writetcfg** parameter.
- Support of alternative groups for DB2 for Linux, UNIX, and Windows servers and DB2 for z/OS servers. See Alternate groups for connections to DB2 Database for Linux, UNIX, and Windows from non-Java clients.

FP6: DB2Connection enhancements

In Version 9.7 Fix Pack 6 and later fix packs, the following features are added to the DB2Connection class:

- Support for the caching of *USRLIBL for connections to DB2 for i V6R1 and later with **CacheUSRLIBLValue** property. See DB2Connection.CacheUSRLIBLValue PropertyDB2Connection.CacheUSRLIBLValue Property.
- Support for clearing of the *USRLIBL cache for connections to DB2 for i V6R1 and later with **ClearUSRLIBLCache** method. See DB2Connection.ClearUSRLIBLCache MethodDB2Connection.ClearUSRLIBLCache Method.
- The default value is provided for the DB2Connection.ClientWorkStation property using the host name. See DB2Connection.ClientWorkStation PropertyDB2Connection.ClientWorkStation Property.

- Support for **DelimIdent** keyword to control whether the connected Informix database server supports delimited SQL identifiers. See `DB2Connection.ConnectionString` Property `DB2Connection.ConnectionString` Property.

FP6: DB2ConnectionStringBuilder enhancements

In Version 9.7 Fix Pack 6 and later fix packs, the following properties are added to the `DB2ConnectionStringBuilder` class:

- Support for changing a user's password with the `DB2ConnectionStringBuilder.NewPWD` property. See `DB2ConnectionStringBuilder.NewPWD` Property `DB2ConnectionStringBuilder.NewPWD` Property.
- Support for setting the CURRENT SQLID special register on DB2 for z/OS by using the new **CurrentSQLID** Data Server Configuration keyword or `DB2ConnectionStringBuilder.CurrentSQLID` property. See `DB2ConnectionStringBuilder.CurrentSQLID` Property `DB2ConnectionStringBuilder.CurrentSQLID` Property.
- The **ZOSDBNameFilter** Data Server Configuration keyword or you can use the `DB2ConnectionStringBuilder.DBName` property to filter the query result of DB2 for z/OS base tables. See `DB2ConnectionStringBuilder.DBName` Property `DB2ConnectionStringBuilder.DBName` Property.

FP6: Data type support enhancements

In Version 9.7 Fix Pack 6 and later fix packs, the following data type enhancements are added:

- SQL data types `SQL_BINARY` and `SQL_VARBINARY` are now supported with DB2 for i V6R1 and later. See SQL data type representation in ADO.NET database applications SQL data type representation in ADO.NET database applications.
- You can now specify XML data type when creating global variables, specifying parameters to create compiled SQL functions, or defining local XML variables in compiled SQL functions. See "FP6: XML data type support added in global variables and compiled SQL functions" on page 26 FP6: XML data type support added in global variables and compiled SQL functions..
- The `DB2Decimal` structure supports the following new fields:

- E
- MinusOne
- One
- Pi
- Zero

See `DB2Decimal` Members `DB2Decimal` Members

- `DB2Blob` and `DB2Clob` supports the following new properties:
 - `EstimatedSize`
 - `IsOpen`
 - `Size`
- `DB2Blob` and `DB2Clob` supports the following new methods:
 - `Read(byte[] buff)`
 - `Read(byte[] buff, Int64 byteOffset, Int64 numBytesToRead, Int64 smartLobOffset, DB2SmartLOBWhence whence)`

- DB2Blob supports DB2Bob(DB2Connection conn) constructor.
- DB2Clob supports DB2Cob(DB2Connection conn) constructor.

FP6: FetchBufferSize keyword support

In Version 9.7 Fix Pack 6 and later fix packs, IBM Data Server Provider for .NET supports setting the **FetchBufferSize** keyword to configure the buffer size used by fetch requests. For more information, see FetchBufferSize IBM Data Server Driver configuration keywordFetchBufferSize IBM Data Server Driver configuration keyword.

FP6: Enhancements to IBM entity provider

In Version 9.7 Fix Pack 6 and later fix packs, IBM entity provider supports the following DB2 and Informix functions for LINQ to Entities queries:

- Acos
- Asin
- Atan
- Atan2
- Cos
- Exp
- Log
- Log10
- Sin
- SquareRoot
- Tan

See Provider support for Microsoft Entity FrameworkProvider support for Microsoft Entity Framework.

FP6: Anonymous block support enhancements

In Version 9.7 Fix Pack 6 and later fix packs, .NET data provider supports the retrieval of the result sets from execution of anonymous blocks by using DB2DataReaderor DB2ResultSet classes.

FP6: Support for Informix data types

In Version 9.7 Fix Pack 6 and later fix packs, .NET data provider supports the following Informix specific features:

- The use of the DB2DateTime structure with Informix database server. See DB2DateTime StructureDB2DateTime Structure.
- New methods GetDB2DateTime() and SetDB2DateTime() for use with the DB2DateTime structure.
- The DB2Type enumeration supports Informix data types for use with Informix database server. See DB2Type EnumerationDB2Type Enumeration.

FP6: DIAGLEVEL setting support in the db2dsdriver.cfg file

In Version 9.7 Fix Pack 6 and later fix packs, you can use the **Diaglevel** keyword in the db2dsdriver.cfg file to configure the diagnostic error capture level configuration parameter. See Diaglevel IBM Data Server Driver configuration keywordDiaglevel IBM Data Server Driver configuration keyword

Other data server support enhancements

IBM Data Server Provider for .NET works with multiple types of IBM data servers. Version 9.7 includes enhancements that improve the performance of .NET applications that connect to DB2 for z/OS and, IBM Informix,

DB2 for z/OS enhancements specific to:

Seamless failover in XA support

Seamless failover improves XA connection reliability for data server applications. The IBM Data Server Provider for .NET supports seamless failover if you have it set up on your data servers.

BinaryXML support

In Version 9.7 Fix Pack 3 and later fix packs, when working with XML columns in DB2 for z/OS Version 10, you can optionally insert and retrieve the XML columns in a binary format that allows it to be processed as a binary object.

Currently committed cursor stability support

In Version 9.7 Fix Pack 3 and later fix packs, you can use a new connection string parameter and db2dsdriver configuration parameter, `ConcurrentAccessResolution`, to use currently committed cursor stability.

Variable length Timestamp support added for DB2 for z/OS data servers

In Version 9.7 Fix Pack 3 and later fix packs, variable length timestamp support in the data provider is extended to DB2 for z/OS data servers.

Connection String Property support

The IBM Data Server Provider for .NET supports a set of connection string properties that provides client information.

IBM Informix enhancements specific to:

ReturnValue parameter support for stored procedures

Informix data server stored procedures can return single or multiple result sets. Previously, IBM Data Server Provider for .NET did not support multiple values from user-defined routines (UDRs). The added support for **ReturnValue** parameters means that IBM Data Server Provider for .NET can retrieve the result set as a single return value.

BIGINT and BIGSERIAL data types support

IBM Data Server Provider for .NET previously supported only INT8 and SERIAL8 data types for 64-bit integers. Support is added for BIGINT and BIGSERIAL data types, which have better performance than INT8 and SERIAL8 data types.

High availability disaster recovery (HADR) support

HADR protects against data loss by replicating data to a secondary database. IBM Data Server Provider for .NET works with this feature if you set it up on your data servers.

Workload Manager (WLM) support

You can use the WLM feature to maximize the use of your resources. IBM Data Server Provider for .NET works with this feature if you set it on your data servers.

FP1: IfxType.Money

Starting in Version 9.7 Fix Pack 1, support for the Informix MONEY data type is provided as an IfxType enumeration. The MONEY data type is treated as a DECIMAL data type with 2 digits of precision.

FP1: Informix optimizer directives

The IBM Data Server Provider for .NET does not process Informix optimizer directives. Starting in Version 9.7 Fix Pack 1, IBM Data Server Provider for .NET passes the directives through the client-side parsing to the data server where all directive-driven optimizations occur.

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

Related reference:

"SQL data type representation in ADO.NET database applications" 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 170)
- User-defined functions support (see “SQL PL functionality has been extended for user-defined functions” on page 171)
- Triggers support (see “Triggers support has been extended” on page 171)
- 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 173)
- Global variable assignments in nested contexts support (see “FP1: Global variable assignments in nested contexts are supported” on page 173)

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 124.

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 127

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
- Assignment to global variables (available in Fix Pack 1 and later fix packs)

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 and in Version 9.7 Fix Pack 1, 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
- OUT and INOUT parameters (available in Fix Pack 1 and later fix packs)
- Compiled UDFs which contain assignment to global variables (available in Fix Pack 1 and later fix packs)

In releases prior to Version 9.7, these features are either not available or are only available for use within SQL procedures.

In Version 9.7 Fix Pack 1 and later fix packs, basic support for compiled user defined functions has been added for database partitioning environments.

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 11. Comparison of SQL syntax required for inline SQL functions and compiled SQL functions

Inline SQL function definition	Compiled SQL function definition
<pre>CREATE FUNCTION TAN (X DOUBLE) RETURNS DOUBLE LANGUAGE SQL CONTAINS SQL NO EXTERNAL ACTION DETERMINISTIC BEGIN ATOMIC RETURN SIN(X)/COS(X); END</pre>	<pre>CREATE FUNCTION TAN (X DOUBLE) RETURNS DOUBLE LANGUAGE SQL CONTAINS SQL NO EXTERNAL ACTION DETERMINISTIC BEGIN RETURN SIN(X)/COS(X); END</pre>

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:

"CREATE FUNCTION (external scalar) " in SQL Reference, Volume 2

FP1: Global variable assignments in nested contexts are supported

Starting in Version 9.7 Fix Pack 1, you can nest global variable assignments in compiled user-defined functions (UDFs) and compiled triggers.

For example, a trigger activated by an INSERT statement can update a global variable.

Related reference:

"Global variables" in SQL Reference, Volume 1

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. DB2 Text Search and Net Search Extender enhancements

Version 9.7 includes enhancements that extend the DB2 Text Search and 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” on page 180.

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.

Related concepts:

"Partitioned table support" in Net Search Extender Administration and User's Guide

Related tasks:

"Creating a text index on range partitioned tables" in Net Search Extender Administration and User's Guide

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.

Related concepts:

"Partitioned database support" in Net Search Extender Administration and User's Guide

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.

Related concepts:

"Extended text-maintained staging infrastructure for incremental update" in Net Search Extender Administration and User's Guide

"Incremental update based on integrity processing" in Net Search Extender Administration and User's Guide

FP3: DB2 Text Search and Net Search Extender index coexistence

Starting with DB2 Version 9.7 Fix Pack 3 and later fix packs, DB2 Text Search and Net Search Extender text indexes can coexist on the same table column. A DB2 Text Search text index can now be created on the same column for which a Net Search Extender index already exists.

If a DB2 Text Search index is created on a column which already has an active Net Search Extender index, the new Text Search index will be marked as inactive by default; otherwise the status will be active. Administrative commands can be executed on both types of text indexes regardless of whether they are active or inactive. Some administrative operations like **DROP INDEX** are allowed on invalid indexes whereas others are blocked. Only active indexes are used for text search.

You can use the `SYSPROC.SYSTS_ALTER` stored procedure or the **db2ts ALTER** command to switch the text index status from **ACTIVE** to **INACTIVE** and vice versa. The active DB2 Text Search index will be used if both Text Search and Net Search Extender index are active on the same column. If all indexes are inactive, an error is returned indicating no index was found.

By specifying the **UNILATERAL** option for the activation, a DB2 Text Search index can be set to inactive independent of the status of a Net Search Extender index.

The text index type is determined on query level instead of on the predicate level. All active indexes on different columns must be of the same index type in a query, that is, either of type Text Search or Net Search Extender.

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 184)
- Additional response file support for uninstalling DB2 products (see “Uninstalling using a response file is supported in more situations” on page 185)
- New response file keywords, **UPGRADE_PRIOR_VERSIONS** and **ACS** (see “New response file keywords have been added” on page 185)

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 184.

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 186 and “Product update service support has been extended” on page 186)
- Additional support for the IBM Tivoli System Automation for Multiplatforms (SA MP) Base Component (see “IBM Tivoli System Automation for Multiplatforms (SA MP) support has been improved” on page 188)

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 188)
- **db2updserv** (show product updates command) support (see “Product update service support has been extended” on page 186)
- New commands to manually create or remove the DB2 tool entries (see “Product installation on Linux and UNIX platforms has been enhanced” on page 187)
- **db21s** (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 187)
- Updates to instance commands (see “Product installation on Linux and UNIX platforms has been enhanced” on page 187)
- FP4: Installation fix pack support has been extended (see “FP4: Installation fix pack support has been extended” on page 190)

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 189)
- An option to not back up some files during installation (see “Fix pack installations can require less space (Linux and UNIX)” on page 189)

Product packaging has been enhanced with the following enhancements:

- IBM solidDB® Universal Cache is now bundled with IBM Database Enterprise Developer Edition (see “FP5: IBM Database Enterprise Developer Edition product bundle extended” on page 190)

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 database 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 database products, features, or languages " in Command Reference

New response file keywords have been added

In Version 9.7, you can use new keywords during unattended installations that use response files.

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.

In Version 9.7 Fix Pack 2 and later fix packs, you can use the **ACS** response file keyword file to install or uninstall the DB2 Advanced Copy Services (ACS) component in response file installations of type custom.

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 300

"INTERACTIVE response file keyword has been changed" on page 241

Related reference:

"Response file keywords" in Installing DB2 Servers

Product installation can be validated using db2val command

The new **db2va1** 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 **db2va1** 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 " 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, `sql1lib/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 **db2is** 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

FP3: 32bit IBM Database Add-ins for Visual Studio is now available with the 64-bit DB2 installation images.

32bit IBM Database Add-ins for Visual Studio is now available with the 64-bit DB2 installation images.

In DB2 Version 9.7 Fix Pack 3 and later fix packs, you can start the 32-bit IBM Database Add-ins for Visual Studio installation from the following parts of the DB2 installation wizard:

- The Install a Product section of the DB2 Setup Launchpad.
- The Install Additional Products section of the DB2 Setup Launchpad, when the DB2 installation has completed.

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

IBM Tivoli System Automation for Multiplatforms (SA MP) support has been improved

Version 9.7 includes a new version of IBM Tivoli System Automation for Multiplatforms (SA MP) and SA MP support has been extended to include Solaris SPARC. Version 9.7 Fix Packs include updated versions of SA MP that you can use in environments with Red Hat Enterprise Linux (RHEL) 6, SUSE Linux Enterprise Server (SLES) 11, AIX 7.1, or POWER7[®] systems.

SA MP is installed automatically on Solaris SPARC, Linux and AIX operating systems. On Windows operating systems, SA MP is bundled with the install media but it is not integrated with the DB2 installer.

Related tasks:

"Installing and upgrading the SA MP with the DB2 installer" in Installing DB2 Servers

Related reference:

"Supported software and hardware for IBM Tivoli System Automation for Multiplatforms (SA MP) " in Installing DB2 Servers

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 " 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. On Linux and UNIX operating systems, to upgrade a single product or to install a product in a new path, use a product-specific fix pack. For Windows operating systems, you can use the universal image to install DB2 to a new location.

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 database products " in Command Reference

FP3: DB2 Connect Unlimited Edition for System z license activation process has been simplified

Starting in Version 9.7 Fix Pack 3, you can activate the license key for DB2 Connect Unlimited Edition for System z[®] on the DB2 for z/OS subsystem if you use DB2 Connect to connect directly to DB2 on System z.

Details

In previous releases, the license key for DB2 Connect Unlimited Edition for System z was activated at each client computer that required access to a z/OS subsystem.

In Version 9.7 Fix Pack 3 and later fix packs, you can instead activate the license key on only the z/OS subsystem or data sharing group that you want to access. You can activate the license only on a z/OS subsystem or data sharing group for which the product was purchased.

If you want to access System z servers through a DB2 Connect gateway server, the license activation process is the same as in previous releases. For more information, see the topics about registering DB2 Connect license keys and setting the license type.

Related tasks:

"Registering a DB2 product or feature license key using the db2licm command" in Installing and Configuring DB2 Connect Servers

"Setting the DB2 license policy using the db2licm command" in Installing and Configuring DB2 Connect Servers

FP4: Installation fix pack support has been extended

In Version 9.7 Fix Pack 4 and higher fix packs, the `installFixPack` command has been enhanced.

New parameter `-f ha_standby_ignore` bypasses checking of the `sql11b` directory. This parameter forces the `installFixPack` command to bypass checking the `sql11b` directory.

On Linux and UNIX platforms, the `installFixPack` command updates the installed DB2 database products in a given location to the same level as the image.

Related reference:

"installFixPack - Update installed DB2 database products " in Command Reference

FP5: IBM Database Enterprise Developer Edition product bundle extended

In Version 9.7 Fix Pack 5 and later fix packs, the IBM Database Enterprise Developer Edition product bundle includes IBM solidDB Universal Cache.

IBM Database Enterprise Developer Edition is not a single product; it is a product bundle. The Database Enterprise Developer Edition bundle is used for high-end development and testing. The list of products included in this bundle now includes solidDB Universal Cache.

Chapter 15. Multicultural support enhancements

Version 9.7 provide more options for working with multicultural data.

The following enhancements are included:

- GB18030 code set support has been extended (see “GB18030 code set support has been extended”)

GB18030 code set support has been extended

Starting with DB2 V9.7 Fix Pack 1, code page 1392 (GB18030) is supported as a client and database code page. Prior to this release, code page 1392 could only be used with the EXPORT, IMPORT, and LOAD utilities with a Unicode database.

To create a database with GB18030 code set, use the following command:

```
CREATE DATABASE ... USING CODESET GB18030 TERRITORY CN
```

You can connect to databases with GB18030 data from clients that either use code page 1392 or Unicode code page 1208 as the application code page.

Windows operating systems do not have a locale setting that reports GB18030 as the code set. To ensure that a DB2 client treats a Windows workstation as using the GB18030 code set (code page 1392), complete the following tasks:

- Install the GB18030 Support Package, which is available from Microsoft.
- In the Regional and Language Options, set the Language for non-Unicode programs setting to Chinese PRC.
- Set the **DB2CODEPAGE** registry variable to 1392.

File names with characters in the GB18030 code set but not in the GBK code set are not supported in the deprecated Control Center. To open or save these files, use CLP commands or CLI.

Related concepts:

"Derivation of code page values" in Globalization Guide

Chapter 16. Troubleshooting and problem determination enhancements

Version 9.7 provides enhancements that make it easier to troubleshoot problems in DB2 environments.

The following enhancements are included:

FP5: Diagnosing upgrade problems is easier

Troubleshooting problems that arise from unsuccessful upgrades is now made easier by support for the collection of diagnostic data before the upgrade operation. You can collect data before upgrading by specifying the new **-preupgrade** parameter for the **db2fodc** command and the **db2support** command. You can collect additional data after the upgrade operation to help troubleshoot an instance creation problem by specifying the new **-c1p** parameter for the **db2fodc** command.

Data that the **-preupgrade** parameters collect provide insight into the system environment and data server software as they were before the upgrade operation.

To collect the data using the **-preupgrade** parameter, first issue the **db2fodc -preupgrade** command. Next, issue the **db2support -preupgrade** command to collect the data from the **db2fodc -preupgrade** command and store it in the `db2support_preupgrade.zip` archive file. If a post-upgrade problem that requires opening a problem management record (PMR) occurs, you can send the `db2support_preupgrade.zip` file to IBM technical support to make the troubleshooting process easier.

To diagnose a problem that occurred during an instance creation, you can use the **-c1p** parameter by issuing the **db2fodc -c1p** command. This command quickly collects environment and configuration related information and stores it in a new directory created under the current diagnostic path, or under a path that you specify. This information makes troubleshooting the instance creation problem easier.

Related reference:

"db2support - Problem analysis and environment collection tool " in Command Reference

"db2fodc - DB2 first occurrence data collection " in Command Reference

FP5: Table space modification status can be checked

In Version 9.7 Fix Pack 5 and later fix packs, the **db2pd -tablespaces** command and the `MON_GET_TABLESPACE` table function provide information about the modification status of table spaces. You can use this information to make better decisions about how you perform backups.

You can now specify the **trackmodstate** option for the **db2pd -tablespaces** command to display the status of the table space with respect to the last backup. In the output, a new `TrackmodState` column is displayed, which can have one of six values for each table space: `Clean`, `Dirty`, `Incremental`, `ReadFull`, `ReadIncremental`, and `n/a`.

The MON_GET_TABLESPACE table function is updated with a new monitor element. This new monitor element is named **tbsp_trackmod_state**. The **tbsp_trackmod_state** monitor element states what status the tablespace is in by displaying one of the six values mentioned previously, except for n/a which is replaced by UNAVAILABLE for the new monitor element.

To receive information about the modification status of table spaces, you must set the **trackmod** configuration parameter to Yes.

Related reference:

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

"tbsp_trackmod_state - Table space trackmod state monitor element" in Database Monitoring Guide and Reference

FP5: db2trc command has been improved for clients

In Version 9.7 Fix Pack 5 and later fix packs, the ability to allocate trace resources for the trace facility (which you invoke with **db2trc** command) helps improve the performance of some remote client applications. Also, clients who use a fenced procedure can capture detailed information about the environment.

For Linux, Solaris, and HP-UX operating systems only, the new **db2trcStartupSize** configuration file keyword is added to help allocate resources for the trace facility on remote clients. The new configuration file keyword must be placed in the `db2dsdriver.cfg` configuration file under the global parameters section to allocate and associate trace resources automatically for remote client applications. When the trace facility is initialized it helps improve the performance of applications running on remote DB2 clients.

The **db2trc** command is also updated to help clients who use a fenced procedure to collect trace related information about the fenced procedure. If you specify the **db2trc** command with either the **-appid** or **-apphdl** parameter, a trace is performed as usual but now it also gathers trace information about fenced procedures.

Note: Modifying the `db2dsdriver.cfg` configuration file or issuing the **db2trc** command must only be done when directed by a DB2 technical support representative.

Related reference:

"db2trc - Trace " in Command Reference

FP5: First occurrence data collection supports new collection types and collection triggered by user-defined thresholds

First occurrence data capture (FODC) collects diagnostic information about your DB2 data server when a problem occurs. In Version 9.7 Fix Pack 5 and later fix packs, FODC supports additional manual collection types and supports triggering automatic diagnostic data collection when a user-defined threshold condition is exceeded.

You can invoke FODC manually with the **db2fodc** command whenever you suspect a problem, or FODC can be invoked automatically whenever a predetermined scenario is detected. For manual FODC, the **-hang** and **-perf** parameters, which have been available since before Fix Pack 5, collect end-to-end diagnostic data. However, these parameters often collect more diagnostic data than is required for troubleshooting, at the cost of additional processor usage and disk space

requirements. On a system that is already resource constrained, any additional demand on resources might not be acceptable, even if it is required to collect diagnostic data. The following new FODC collection types, which apply to more specific performance problem scenarios, help address the problem of additional overhead:

-cpu

If you observe unusually high processor utilization rates, a high number of running processes, or high processor wait times, you can use the **-cpu** parameter to collect processor-related performance and diagnostic data.

-memory

If you determine that there is no free memory available, that swap space is being used at a high rate, or that excessive paging is occurring or if you suspect a memory leak, you can use the **-memory** parameter to collect memory-related diagnostic data.

-connections

If you determine that there is a spike in the number of applications in the executing or compiling state or that new database connections are being denied, you can use the **-connections** parameter to collect connection-related diagnostic data.

With the new **-detect** parameter for the **db2fodc** command, you can now specify your own threshold rule for a specific condition and trigger diagnostic data collection when the condition is exceeded. The **-detect** parameter supports detecting trigger conditions one or several times, at a regular interval that you specify. If the number of times that the threshold condition is detected matches the specified value, diagnostic data collection is triggered. Other options for detecting threshold conditions are also available, such as how many iterations of threshold detection and diagnostic data collection are performed, and how long threshold detection continues.

You can also configure the new user-defined thresholds to only detect and a specific problem condition but not collect diagnostic information. If a problem condition is detected, only a log record is added to the db2diag log files.

Related concepts:

"Collecting diagnostic information based on common outage problems" in Troubleshooting and Tuning Database Performance

Related reference:

"db2fodc - DB2 first occurrence data collection " in Command Reference

FP5: Load serviceability has been improved

In Version 9.7 Fix Pack 5 and later fix packs, load operations are easier to troubleshoot because additional diagnostic information is available through the **db2pd** command. You can obtain the diagnostic information by using the enhanced **-utilities** parameter and the new **-load** parameter.

The enhanced **db2pd -utilities** output includes the load ID and the application ID in the Description column. You can use the load ID to identify the corresponding log entries for a load operation in the db2diag log files.

The new **-load** parameter displays engine dispatchable unit (EDU) information such as the EDU name, EDU ID, application handle, application ID, load ID, load start time, and load phase for all load operations. You can use the EDU

information obtained to perform further troubleshooting steps on problematic load operations, such as running the **db2trc** command. The **-load** parameter has three options:

- The **loadID** option returns all EDU information for a specific load operation.
- The **file** option redirects the output to a specified file.
- The **stacks** option dumps the stack traces for the load EDUs that are stored in the **diagpath** directory.

Related reference:

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

FP5: db2dart command has extended functionality that helps improve performance

In Version 9.7 Fix Pack 5 and later fix packs, the **db2dart** command is updated with extended functionality for the **/T**, **/TSC**, and **/TS** actions and **/QCK**, **/OI** and **/TSI** options that helps to improve the performance of the command.

The **/QCK** option is updated to take in numeric values that perform a different operation for each value. There are 4 main bit-value quick options (1, 2, 4, and 8) that can be added together to perform multiple operations. The quick options skip certain steps when the **db2dart** command examines databases. Skipping unnecessary steps helps improve the performance of the **db2dart** command.

You can now specify a list of table object IDs for the **/T** parameter and **/OI** option and a list of table space IDs for the **/TSC** and **/TS** parameters and the **/TSI** option. This helps improve performance when inspecting tables.

Related reference:

"db2dart - Database analysis and reporting tool " in Command Reference

FP4: The serviceability of large database systems has improved

In Version 9.7 Fix Pack 4 and later fix packs, the serviceability of large database systems has improved with new troubleshooting functionality that better meets the demands of large database environments.

Serviceability pain points

DB2 troubleshooting tools provide highly granular access to diagnostic data that is used to resolve problems on your data server. In large database environments, the collection of diagnostic data can introduce an unwanted impact to the database environment because of the following issues:

- The large volume of diagnostic data generated on file systems and the challenge of sending this volume of data to IBM for analysis
- The impact that diagnostic data collection has on the performance of database systems and the difficulty of selectively collecting diagnostic data

A number of improvements to DB2 troubleshooting tools address these pain points:

First-occurrence data capture (FODC) member-level settings and FODC redirection

The implementation of first-occurrence data capture changed so that each member in the database system can now have its own FODC settings. Member-level FODC settings give you greater control than the

instance-level or host-level settings supported in previous releases and fix packs. As a result, it is now easier to locate the diagnostic information for a specific member in the database environment, or to run multiple automatic or manual FODC processes in parallel. For example, you can now collect diagnostic data only from a specific member that encounters a problem and not have diagnostic data from other members on the same host included.

When errors occur, the automatic capture of important diagnostic data can generate a significant volume of diagnostic data that requires space on the file system to store. To avoid a scenario where first-occurrence data capture fills all the available space in the file system and affects your data server, you can specify where the FODC data is stored with the **FODCPATH** registry variable.

Locally installed support tools and option to extract the `db2support` command package

Several tools are used frequently by IBM technical support service analysts to diagnose data server problems, but in past releases and fix packs these tools were not provided with the product. To save time during the diagnosis of problems, when a problem cannot be reproduced easily at IBM, or when sending a large volume of diagnostic data to IBM is not feasible, these support tools are now available to IBM service analysts locally in your product installation.

The **db2support** command collects diagnostic data in a single compressed package for transmission to IBM technical support. The command now supports a new **-unzip** option to extract the **db2support** command package locally. Together, the locally installed support tools and the support to extract the **db2support** command package make it possible for service analysts to perform problem diagnosis at your site without requiring any tools or diagnostic data to be sent back and forth between IBM and you. The **-unzip** option also makes the **db2support** command a more comprehensive tool for working with archived diagnostic data because you can now use the command both to archive the diagnostic data and to extract from archived diagnostic data without the need for additional software.

Improved granularity when collecting diagnostic data

To avoid the impact of unnecessary diagnostic data collection in large database environments, several troubleshooting commands support new options to control the granularity of diagnostic data collection. These new options speed up data collection by collecting only relevant information, which reduces the performance impact of data collection on the system and can shorten the time required to perform problem determination by IBM technical support.

Simplified syntax for global diagnostic data collection

Collecting diagnostic information globally previously required the **-global** option to be specified for the various diagnostic tool commands, even when remote hosts and partitions were also specified. To simplify the syntax, the mandatory use of the **-global** option on remote hosts and partitions is no longer required. The functionality provided by the **-global** option is deprecated and replaced with the **-member** and **-host** options for the following troubleshooting tools:

- **db2trc**
- **db2pd**
- **db2fodc**

- **db2pdcfg**
- **db2support**

The **-member** option can be used to specify any database partition number, while **-host** is used to specify any host. If you want to collect diagnostic information about all members globally, without having to specify every member in the system, you can use the **-member all** option in place of the deprecated **-global** option. If you want to restrict the collection of diagnostic information to only a specific remote host, which was not possible with the **-global** option, you can use the **-host** option.

Individual troubleshooting improvements

The DB2 troubleshooting commands have the following improvements:

New support scripts: **db2snapcore**, **db2trcon**, and **db2trcoff**

IBM service analysts can use the following new tools during problem diagnosis. These tools are provided with the product.

- On Solaris and Linux operating systems only, the **db2snapcore** command extracts the shared objects list section from the engine dispatchable unit (EDU) trap file and adds them together with the core file to a compressed archive that you can send to DB2 support for analysis. The functionality provided by **db2snapcore** is similar to the **snapcore** command on the AIX operating system.
- The **db2trcon** command turns on the DB2 trace facility for a time period you specify. You can use this script to turn on the trace facility only for top processor time consuming EDUs. You can specify how many engine dispatchable units you want the DB2 trace to be turned on for, and for how long.
- The **db2trcoff** command turns off the DB2 trace facility and generates dump, flow, and format files automatically with a single command.

db2diag command

The **db2diag** command supports a new **-lastrecords** *number-of-records* parameter option. You use this option to output a specific number of diagnostic records added most recently to the **db2diag** log file.

db2pd command

The **db2pd** command supports two new options for the **-edus interval** parameter. The two new options for the **-edus interval** parameter are **top** and **stacks**. The **top** option outputs the top engine dispatchable units based on processor time consumed. The **stacks** option dumps stack information for EDUs returned by the **db2pd** command.

On UNIX and Linux operating systems only, the **db2pd** command also supports two new options for each of the **-dump** and **-stack** parameters. The two new options are **dumpdir** and **timeout**. The **dumpdir** option specifies a directory where stack files are redirected to. Use the **timeout** option to specify time length for the redirection of the stack files to a specific directory.

db2trc command

The **-p** parameter for the **db2trc** command now supports a simplified syntax for specifying multiple thread IDs (*tids*) with a single process ID (*pid*). For example, *pid.tid1.tid2.tid3* is now a valid *pid-tid* pairing, specifying three thread IDs with a single process ID.

db2fodc command and FODCPATH registry variable parameter

Both automatic and manual FODC invocation through the **db2fodc** command now support member-level FODC settings and redirection of FODC packages to a directory path you specify. When FODC is invoked automatically, the new **FODCPATH** parameter for the **DB2FODC** registry variable specifies the path to where FODC packages are stored either persistently or dynamically. Persistent storage is enabled by using the **db2set** command, while dynamic storage, which stores FODC packages in memory until the instance is recycled, is enabled by using the **db2pdcfg** command. When you run **db2fodc** command manually, you can use the new **-fodcpath** parameter to specify a directory path where FODC packages are stored.

db2support command

The **db2support** command supports two new options, the **-fodcpath** and **-unzip** options. The **-unzip** option extracts the diagnostic files from the generated **db2support** package, used where problem diagnosis by a service analyst takes place directly on your system. Use the **-fodcpath** option to specify the path where an FODC package to collect is stored. The **db2support** command also supports the new **FODCPATH** registry variable and collects FODC packages from the path specified.

Related concepts:

"First occurrence data capture information" in Troubleshooting and Tuning Database Performance

"Analyzing db2diag log files using db2diag tool" in Troubleshooting and Tuning Database Performance

"Obtaining a DB2 trace using db2trc" in DB2 Connect User's Guide

"First occurrence data capture configuration" in Troubleshooting and Tuning Database Performance

"Monitoring and troubleshooting using db2pd command" in Troubleshooting and Tuning Database Performance

Related tasks:

"Collecting environment information using db2support command" in Troubleshooting and Tuning Database Performance

Related reference:

"db2set - DB2 profile registry " in Command Reference

"db2trc - Trace " in Command Reference

"db2support - Problem analysis and environment collection tool " in Command Reference

"General registry variables" in Database Administration Concepts and Configuration Reference

"db2diag - db2diag logs analysis tool " in Command Reference

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

"db2pdcfg - Configure DB2 database for problem determination behavior " in Command Reference

"db2fodc - DB2 first occurrence data collection " in Command Reference

"db2snapcore - DB2 snapcore command for Linux and Solaris " in Command Reference

"db2trcon - On trace options for db2trc" in Command Reference

"db2trcoff - Off trace options for db2trc" in Command Reference

FP4: New configuration parameter reduces risk of losing diagnostic data

In Version 9.7 Fix Pack 4 and later fix packs, you can specify an alternative path to a directory where DB2 diagnostic information is stored by using the new **alt_diagpath** database manager configuration parameter.

The **alt_diagpath** database manager configuration parameter is used only if DB2 cannot write to the path set by the **diagpath** database manager configuration parameter.

If DB2 fails to write to the directory that you specify by using the **diagpath** configuration parameter, important diagnostic information might be lost. To reduce the chances of losing diagnostic information, you should set the **alt_diagpath** configuration parameter, which does not have a default value. If you set the **alt_diagpath** configuration parameter to the same path that you use for the **diagpath** configuration parameter, an error message is generated. Also, you should not set the **diagpath** and **alt_diagpath** configuration parameters to the same file system. If you do, a warning message is generated.

Related concepts:

"Diagnostic data directory path" in Troubleshooting and Tuning Database Performance

Related reference:

"alt_diagpath - Alternate diagnostic data directory path " in Database Administration Concepts and Configuration Reference

FP4: Archive log files can be checked for validity

In Version 9.7 Fix Pack 4 and later fix packs, you can check the validity of archive log files with the **db2cklog** command before using these files during a rollforward recovery operation.

The **db2cklog** command reads either a single log file or a range of log files. You can use this command immediately before a rollforward recovery to ensure that the recovery operation will not fail because of a problem with a log file. If the **db2cklog** command returns error messages or warnings for a log file, do not use that file during rollforward recovery without addressing the problems.

If IBM Software Support suspects that an invalid log file is causing a problem with your data server, you might be asked to run the **db2cklog** command. You can also use the command to validate each log file after it is closed and copied to the log archive directory.

Related tasks:

"Checking archive log files with the db2cklog tool" in Troubleshooting and Tuning Database Performance

Related reference:

"db2cklog - Check archive log files command" in Command Reference

FP4: DB2 Text Search infrastructure improvements

Starting with DB2 Version 9.7 Fix Pack 4 and later fix packs, a new Text Search Server release is integrated. The new release provides improved performance, and enables more differentiated configuration and tuning.

This new Text Search server processes text in 22 languages, supports a wide range of encoding, includes text extraction from some binary formats and provides rich text support. It provides stronger indexing and search capabilities on the DB2 server host machine.

You will need to install the new DB2 Accessories Suite to setup rich text document support for DB2 Text Search servers.

FP3: Improvements to the granularity of the db2trc tool

Starting with V9.7 FP3, two new features have been added to improve the granularity of the db2trc tool. These are an ability to trace only the members (or partitions) specified and an ability to trace based on a specific application ID (or application handle).

To provide this functionality the following trace masks have been added to the db2trc tool.

-appid Use this parameter to trace specific application IDs. The **-appid** option works with the **on** and **change** command. **-appid** will not work with the **-perfcount** option

-apphdl

Use this parameter to trace specific application handles. The **-apphdl** option works with the **on** and **change** command. **-apphdl** will not work with the **-perfcount** option

-member

Specifies which database members (or partitions) to trace. The **-member** option works with the **on**, **change**, **format** (both **flow** and **format** options), **stop**, and **off** options.

Related reference:

"db2trc - Trace " in Command Reference

FP1: Diagnostic data can be stored in separate directories

Starting with Version 9.7 Fix Pack 1, you can specify to store DB2 diagnostic data in separate directories named according to the physical host, database partition, or both by setting the enhanced **diagpath** database manager configuration parameter. Separate **db2diag** log files can later be merged together using the **db2diag -merge** command.

The benefits of splitting the diagnostic data into separate directories are as follows:

- Diagnostic logging performance can be improved because of less contentions on the **db2diag** log file if you split the diagnostic data per host or per database partition.
- Storage management can be under more granular control.

To split the diagnostic data into separate directories, set the **diagpath** database manager configuration parameter to one of the following values:

- Split default diagnostic data directory path according to physical host:
`db2 update dbm cfg using diagpath "$h"`
- Split your own specified diagnostic data directory path according to physical host:
`db2 update dbm cfg using diagpath "pathname $h"`
- Split default diagnostic data directory path according to database partition:
`db2 update dbm cfg using diagpath "$n"`
- Split your own specified diagnostic data directory path according to database partition:
`db2 update dbm cfg using diagpath "pathname $n"`
- Split default diagnostic data directory path according to physical host and database partition:
`db2 update dbm cfg using diagpath "hn"`
- Split your own specified diagnostic data directory path according to physical host and database partition:
`db2 update dbm cfg using diagpath "pathname hn"`

Merging separate **db2diag** log files can, at times, make analysis and troubleshooting easier. In that case, you can use the **db2diag -merge** command.

Related concepts:

"Diagnostic data directory path" in Troubleshooting and Tuning Database Performance

Related tasks:

"Splitting a diagnostic data directory path by database partition server, database partition, or both" in Troubleshooting and Tuning Database Performance

Related reference:

"diagpath - Diagnostic data directory path " in Database Administration Concepts and Configuration Reference

"db2diag - db2diag logs analysis tool " in Command Reference

FP1: db2support tool has been enhanced

Starting in Fix Pack 1, the **db2support** tool includes new filtering options that you can use to gather specific diagnostic data more easily and an archiving option for storing diagnostic files in a different location.

You can use the following new options:

- The **-history** *history period* and **-time** *time interval* options limit the data gathered by the **db2support** tool to the history period or time interval that you specify.
- The **-Archive** *archive path* option creates a copy of the contents of the directory specified by the **DIAGPATH** configuration parameter in an archive path that you specify. The archived directory's name is appended with the hostname and current time stamp automatically.
- The **-basic** option limits the data gathered by the **db2support** tool to the optimizer-related diagnostic information.
- The **-o1** option has been enhanced to support gathering data for multiple optimization levels.
- The **-extenddb2batch** option allows **db2batch** information to be collected for all optimization levels when used with the **-o1** and **-c1** options.
- The **-nodb2look** and **-nocatalog** options prevent the collection of db2look information and catalog information respectively.

Related reference:

"db2support - Problem analysis and environment collection tool " in Command Reference

FP1: Fenced routines history information is easier to collect

Starting in DB2 Version 9.7 Fix Pack 1, you can keep track of the execution history of fenced routines (including those that attempted to be loaded) more easily using the output of the **db2pd** command with the new **-fmpexechistory** parameter.

You can use the **-fmpexechistory** parameter to display the execution history of fenced routines (including the routines that attempted to run) in order to diagnose some FMP process related issues.

To help interpret the fenced routines history information provided by the **db2pd** command, you can use the **genquery** option to generate a query that you can save and reuse to return the routine schema, module, name, and specific name according to a routine unique ID. You can run this query after the database

manager is stopped and started, and as long as no routine is dropped, the query result will reflect the routine execution history collected at the point when the **db2pd** command was run.

Related reference:

"db2pd - Monitor and troubleshoot DB2 database " 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.

Note: Starting with DB2 Version 9.7 Fix Pack 1, if the **diagsize** configuration parameter is set to a non-zero value and the **diagpath** configuration parameter is set to split the diagnostic data into separate directories, then the non-zero value of the **diagsize** configuration parameter specifies the total size of the combination of all rotating administration notification log files and all rotating diagnostic log files contained within a given split diagnostic data directory. For example, if a system with 4 database partitions has **diagsize** set to 1 GB and **diagpath** set to "\$n" (split diagnostic data per database partition), the maximum total size of the combined notification and diagnostic logs can reach 4 GB (4 x 1 GB).

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 Troubleshooting and Tuning Database Performance

"FP1: Diagnostic data can be stored in separate directories" on page 202

Related reference:

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

Part 2. 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 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. You can replace DB2 Connect server with DB2 Connect client and receive equivalent or superior function. In addition, you can reduce complexity, improve performance, and deploy application solutions with smaller footprints for your business users.

Version 9.7 enhancements and changes

The following Version 9.7 enhancements and changes affect the functionality and capabilities of DB2 Connect. Topics that are associated with a specific fix pack include an "FPx" prefix at the beginning of the topic title, where x represents a fix pack level.

Product packaging enhancements

- "Component names have changed" on page 3

Security enhancements

- "SSL client support expanded and configuration simplified" on page 101
- "AES encryption of user ID and password enhances security" on page 100
- "FP1: Transparent LDAP authentication and group lookup is supported (Linux and UNIX)" on page 103
- "FP1: 32-bit GSKit libraries are included in the 64-bit DB2 product installation" on page 105

Application development enhancements

- "Common SQL API supports the development of portable administrative applications" on page 125
- "Python application development support has been added" on page 123
- "IBM Database Add-Ins for Visual Studio have been enhanced" on page 126

DB2 Connect client enhancements

- "JDBC and SQLJ support has been enhanced" on page 132
- "IBM Data Server Driver Package has been enhanced" on page 148

- “IBM Data Server Provider for .NET is enhanced” on page 160
- “Trusted context support has been extended” on page 149
- “Sysplex support is extended to IBM data server clients and non-Java data server drivers” on page 149
- “Call level interface (CLI) functionality has been enhanced” on page 150

High availability, backup, logging, resiliency, and recovery enhancements

- “Enhanced resilience to errors and traps reduces outages” on page 58

Installation, upgrade, and fix pack enhancements

- “Instances and DB2 Administration Server can be created in a shared DB2 copy (Linux and UNIX)” on page 184
- “Installation images can be reduced (Linux and UNIX)” on page 188
- “Product installation can be validated using db2val command” on page 186
- “Product update service support has been extended” on page 186
- “Product installation on Linux and UNIX platforms has been enhanced” on page 187
- “Universal fix packs support has been extended (Windows)” on page 189
- “Fix pack installations can require less space (Linux and UNIX)” on page 189
- “Uninstalling using a response file is supported in more situations” on page 185
- “New response file keywords have been added” on page 185
- “db2rspgn command supported on Linux and UNIX operating systems” on page 184
- “FP4: Installation fix pack support has been extended” on page 190

Licensing changes

- “FP3: DB2 Connect Unlimited Edition for System z license activation process has been simplified” on page 189

Multicultural support enhancements

- “GB18030 code set support has been extended” on page 191

Troubleshooting and problem determination enhancements

- “FP1: Diagnostic data can be stored in separate directories” on page 202
- “FP1: db2support tool has been enhanced” on page 203
- “Administration notification and diagnostic logs occupy specified amount of disk space” on page 60

Administration changes

- “Some database manager configuration parameters have been changed” on page 222
- “Some registry and environment variables have changed” on page 225
- “DESCRIBE command lists information about additional index types” on page 233
- “Registry files have been removed from the DB2 installation path” on page 241

Security changes

- “System administrator (SYSADM) authority scope has changed” on page 243
- “Security administrator (SECADM) abilities have been extended” on page 245
- “Database administrator (DBADM) authority scope has changed” on page 246
- “SSLconfig.ini and SSLClientconfig.ini files replaced with new database manager configuration parameters” on page 247

Application development changes

- “Merge modules for ODBC, CLI, and .NET have been combined (Windows)” on page 271

Deprecated functionality

- “db2iupdt command -s option has been deprecated (Linux and UNIX)” on page 299
- “Control Center tools have been deprecated” on page 276
- “Health monitor has been deprecated” on page 288
- “Some response file keywords have been deprecated” on page 300
- “Instance and database migration commands and APIs have been deprecated” on page 299
- “Worksheet Format (WSF) for Import and Export utilities has been deprecated” on page 291
- “FP3: HP-UX 32-bit client support has been deprecated” on page 297

Discontinued functionality

- “db2secv82 command has been discontinued” on page 310
- “db2ilist command options -a and -p have been discontinued” on page 311
- “Netscape browser support has been discontinued” on page 307
- “Some operating systems are no longer supported” on page 305

Chapter 17. DB2 Connect Version 9.7 fix pack summary

DB2 Version 9.7 fix packs includes important changes to existing features and additional features that might affect DB2 Connect usage.

If you did not apply the Version 9.7 fix packs or did not update your local Information Center since Version 9.7 was available, you should review the following topics to understand the technical changes included in the Version 9.7 fix packs that might affect DB2 Connect. Fix packs are cumulative, they contain all of the changes and functionality shipped in previous fix packs.

- Fix Pack 6
- Fix Pack 5
- Fix Pack 4
- “Fix Pack 3” on page 214
- “Fix Pack 2” on page 214
- “Fix Pack 1” on page 215

Fix Pack 6

Fix Pack 6 contains the functionality of previous fix packs and includes the following changes:

- The behavior of the DB2 call level interface (CLI) is changed when you specify both the **SchemaList** and **SchemaFilter** keywords with *USRLIBL value for connections to DB2 for i. For more information, see “SchemaFilter IBM Data Server Driver configuration keyword” and “SchemaList CLI/ODBC configuration keyword”.
- The default values of selected data server driver keyword parameters that are associated with automatic client reroute have changed. For more information, see “Default values of selected automatic client reroute IBM Data Server Driver keywords have changed” on page 252.
- The default value or behavior of selected data server driver keywords have changed. For more information, see “Default value or behavior of selected IBM Data Server Driver keywords have changed” on page 252.
- The behavior of IBM .NET Data Provider is changed if you specify both the **SchemaFilter** keyword and the `DB2ConnectionStringBuilder.SchemaList` property with *USRLIBL value for connections to DB2 for i. For more information, see “SchemaFilter IBM Data Server Driver configuration keyword” and “`DB2ConnectionStringBuilder.SchemaList` Property”.

Fix Pack 6 contains the functionality of previous fix packs and includes the following enhancements:

- Versions of the IBM Data Server Driver for JDBC and SQLJ that are shipped with this fix pack contain several enhancements. For more information, see “FP6: Driver versions enhancements” on page 147.
- The **db2mcs** command now provides the **-user** and **-passwd** parameters to avoid specifying this information in the `db2mcs.cfg` file. For more information, see “db2mcs - Set up Windows failover utility command”.
- The `SQL_BINARY` and `SQL_VARBINARY` SQL data types are now supported with DB2 for i V6R1 and later. For more information, see “SQL data type representation in ADO.NET database applications”.

- CLI now supports SQL BINARY and SQL VARBINARY data types for DB2 for i Version 6 Release 1 and later. For more information, see “SQL symbolic and default data types for CLI applications”.
- CLI now supports array input by using the SQL_ATTR_PARAMSET_SIZE statement attribute, inside a trusted procedure body. For more information, see “Statement attributes (CLI) list”.
- CLI now supports retrieval of the server time for COMMIT or ROLLBACK SQL operations on DB2 for z/OS Version 10 and later. For more information, see “Connection attributes (CLI) list”.
- CLI now supports the SQL_C_CURSORHANDLE C data type for use with the SQL_CURSORHANDLE SQL data type. For more information, see “SQL symbolic and default data types for CLI applications”.
- CLI now supports the **QueryTimeout** keyword in the db2dsdriver.cfg file. For more information, see “QueryTimeout IBM Data Server Driver configuration keyword”.
- You can specify LDAP variables using the data server driver keywords. For more information, see “Call level interface (CLI) functionality has been enhanced” on page 150.
- Windows system environment variables are now updated to include the IBM Data Server Driver Package path. For more information, see “FP6: Environment variables now contain the driver path (Windows)” on page 131“FP6: Environment variables now contain the driver path (Windows)”.
- You can now check the status of NSE text indexes. For more information, see “FP6: New utility validates a Net Search Extender text index (Windows, AIX)” on page 131“FP6: Validate your Net Search Extender text index”.
- You can use the **SQLCODEMAP** Data Server Driver configuration keyword or the **SQLCODEMAP** CLI/ODBC keyword to specify whether SQLCODE mapping should be set. For more information, see “SQLCODEMAP IBM Data Server Driver Configuration Keyword”.
- IBM .NET Data Provider now supports changing a user password by using the DB2ConnectionStringBuilder.NewPWD property. For more information, see “DB2ConnectionStringBuilder.NewPWD Property”.
- IBM .NET Data Provider now supports setting the CURRENT SQLID special register on DB2 for z/OS servers by using the new **CurrentSQLID** IBM Data Server Driver configuration keyword or the DB2ConnectionStringBuilder.CurrentSQLID property. For more information, see “DB2ConnectionStringBuilder.CurrentSQLID Property”.
- IBM .NET Data Provider now supports setting the **FetchBufferSize** IBM Data Server Driver configuration keyword to configure the buffer size that is used by fetch requests. For more information, see “FetchBufferSize IBM Data Server Driver configuration keyword”.
- You can use the **ZOSDBNameFilter** IBM Data Server Driver configuration keyword or DB2ConnectionStringBuilder.DBName property to filter the query result of DB2 for z/OS base tables. For more information, see “DB2ConnectionStringBuilder.DBName Property”.
- You can use the CacheUSRLIBLValue property to cache *USRLIBL for connections to DB2 for i V6R1 and later. For more information, see “DB2Connection.CacheUSRLIBLValue Property”.
- IBM .NET Data Provider now supports **DelimIdent** keyword and provides default value for the ClientWorkStation property. For more information, see “IBM Data Server Provider for .NET is enhanced” on page 160“IBM Data Server Provider for .NET has been enhanced”.

- IBM .NET Data Provider now supports DB2DateTime structure for use with Informix database server. For more information, see DB2DateTime StructureDB2Date Time Structure.
- IBM entity provider supports DB2 and Informix functions for LINQ to Entities queries. For more information, see Provider support for Microsoft Entity FrameworkProvider support for Microsoft Entity Framework.
- New fields are added to the DB2Decimal class. For more information, see “IBM Data Server Provider for .NET is enhanced” on page 160“IBM Data Server Provider for .NET has been enhanced”.
- The db2dsdriver.cfg file can be used to set DIAGLEVEL. For more information, see Diaglevel IBM Data Server Driver configuration keywordDiaglevel IBM Data Server Driver configuration keyword.
- New properties and methods are added to the DB2Blob and DB2Clob classes. For more information, see “IBM Data Server Provider for .NET is enhanced” on page 160“IBM Data Server Provider for .NET has been enhanced”.
- You can use the ClearUSRLIBLCache method to clear the *USRLIBL cache for connections to DB2 for i V6R1 and later. For more information, see “DB2Connection.ClearUSRLIBLCache Method”.
- Enhancements to the **db2cli** command include new features to the **registerdsn** parameters in Windows. For more information, see “Call level interface (CLI) functionality has been enhanced” on page 150.
- The **installDSDriver** command on UNIX and Linux operating systems now creates the db2profile and db2cshrc files to set required environment variables. For more information, see “installDSDriver - Extract Data Server Driver components command”.
- Seamless automatic client reroute is enhanced. The CLI driver can now perform seamless failover when a COMMIT or ROLLBACK statement is issued against an unreachable server after all the data is returned. For more information, see “Call level interface (CLI) functionality has been enhanced” on page 150.
- Enhancements to CLI include support for client information properties against DB2 for i. For more information, see “Call level interface (CLI) functionality has been enhanced” on page 150.
- Embedded SQL applications can use timeout values specified in the db2dsdriver.cfg file. For more information, see “Use of the db2dsdriver.cfg configuration file by embedded SQL application”.
- Embedded SQL applications can call stored procedure using three-part name. For more information, see “Enabling compatibility features for migration”.
- Embedded SQL applications can call stored procedure using the new syntax. For more information, see “Enabling compatibility features for migration”.
- Embedded SQL applications can use **WHENEVER condition DO action** statement to take a specified action when an exception condition occurs. For more information, see “WHENEVER statement ”.

Fix Pack 5

Fix Pack 5 contains the functionality of previous fix packs and includes the following change:

- The IBM Software Development Kit for Java that is packaged with DB2 products now use the Java 6.0.9.1 version. This Java version is now the minimum level supported by DB2 products. It addresses a critical security vulnerability (CVE-2010-4476) that might cause the Java Runtime Environment to hang. For

more information about Java versions that are bundled with DB2 products, see “Java software support for DB2 products”.

Fix Pack 5 also contains the following enhancements:

- Merged backup images are now fully integrated with automatic recovery utilities and database history management infrastructure. For more information, see “FP5: Automatic recovery utilities now recognize merged backups” on page 63.
- You can now use the new `SQLADM_NO_RUNSTATS_REORG` option of the `DB2AUTH` registry variable to specify that users with `SQLADM` authority cannot issue `runstats` or `reorg` operations. For more information, see “DB2 authorization model has been enhanced to allow separation of duties” on page 98.

Fix Pack 4

Fix Pack 4 contains the functionality of previous fix packs and includes the following change:

- Support to perform distributed installations using Microsoft Systems Management Server on Windows operating systems is deprecated. For more information, see “FP4: Distributed installation support with Microsoft Systems Management Server is deprecated (Windows)” on page 301.

Fix Pack 4 also contains the following enhancements:

- You can use the `-recovery` option of the `db2pd` command to determine whether the catalog database partition has failed. For more information, see “`db2pd - Monitor and troubleshoot DB2 database` command” in the *Command Reference*.
- New `installFixPack` command parameter enhances ability to apply fix packs. For details, see “FP4: Installation fix pack support has been extended” on page 190.

Fix Pack 3

Fix Pack 3 contains the functionality of previous fix packs and includes the following enhancements:

- Support for the AIX 7.1 operating system. For more information, see “Installation requirements for DB2 Connect server products (AIX)” in *Installing and Configuring DB2 Connect Servers*.
- Two new features have been added to improve the granularity of the `db2trc` tool, these are an ability to trace only the members (or partitions) specified and an ability to trace based on a specific application ID (or application handle). For more information, see “FP3: Improvements to the granularity of the `db2trc` tool” on page 201.

Fix Pack 2

Fix Pack 2 contains the functionality of Fix Pack 1 and includes the following enhancements:

- Support for IBM Rational® Developer for zSeries® v7. For more information see “Support for database application development in COBOL” in *Database Administration Concepts and Configuration Reference*.
- The `db2pd` command has additional functionality to help control which engine dispatchable units (EDUs) are included in the output, the ability to define a time interval in some cases, and improved stack output on Linux operating systems. For more information, see “`db2pd - Monitor and troubleshoot DB2 database`” in *Command Reference*.

- The IBM Data Server Provider for .NET has been enhanced in several ways that could help with application performance, data server compatibility, and simplifying application development. For more information, see “IBM Data Server Provider for .NET is enhanced” on page 160.

Fix Pack 1

Fix Pack 1 includes the following enhancements:

- The IBM Data Server Provider for .NET includes multiple enhancements. For more information, see “IBM Data Server Provider for .NET is enhanced” on page 160.
- The **diagpath** database manager configuration parameter has new values, which allow you to store DB2 diagnostic data in separate directories named according to the physical host, database partition, or both. The **db2diag** command also has a new **-merge** parameter to merge multiple **db2diag** log files. For more information, see “FP1: Diagnostic data can be stored in separate directories” on page 202.
- Transparent LDAP is supported on the Linux, HP-UX, and Solaris operating systems. For more information, see “FP1: Transparent LDAP authentication and group lookup is supported (Linux and UNIX)” on page 103.
- 32-bit GSKit libraries are now installed automatically. For more information, see “FP1: 32-bit GSKit libraries are included in the 64-bit DB2 product installation” on page 105.
- Additional support is provided for the GB18030 code set. For more information, see “GB18030 code set support has been extended” on page 191.
- DB2 database products installed on HP-UX operating systems now support long host names. For more information, see “Installation requirements for DB2 Connect server products (HP-UX)” in SC27-2433-03.

Chapter 18. FP4: Installation fix pack support has been extended

In Version 9.7 Fix Pack 4 and higher fix packs, the `installFixPack` command has been enhanced.

New parameter `-f ha_standby_ignore` bypasses checking of the `sqllib` directory. This parameter forces the `installFixPack` command to bypass checking the `sqllib` directory.

On Linux and UNIX platforms, the `installFixPack` command updates the installed DB2 database products in a given location to the same level as the image.

Related reference:

"installFixPack - Update installed DB2 database products " in Command Reference

Part 3. 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 19, “Changed functionality,” on page 221

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 20, “Deprecated functionality in Version 9.7,” on page 275

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 21, “Discontinued functionality in Version 9.7,” on page 305

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

Chapter 22, “Summary of deprecated and discontinued DB2 functionality in Version 9.7 and earlier releases,” on page 313

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 19. 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 data 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.

In the previous situations, a nonpartitioned index is created by default.

Starting in DB2 Version 9.7 Fix Pack 1, you can create indexes over XML data on a partitioned table as either partitioned or nonpartitioned. The default is a partitioned index.

With DB2 V9.7 and earlier, for tables created that use multidimensional clustering (MDC) and data partitioning, the system-created MDC block indexes are always created as nonpartitioned indexes. Starting in DB2 V9.7 Fix Pack 1, for tables created that use both MDC and data partitioning, the system-created MDC block indexes are always created as partitioned indexes. DB2 V9.7 Fix Pack 1 and later releases support partitioned MDC tables with nonpartitioned block indexes and partitioned MDC tables with partitioned block indexes.

User response

If you do not want to create partitioned indexes on partitioned tables, specify the `NOT PARTITIONED` clause on the `CREATE INDEX` statement.

If you have a data-partitioned MDC table that uses nonpartitioned block indexes, and you want to use partitioned block indexes, you must create new data-partitioned MDC table that uses partitioned block indexes, and move the data in the existing table to the new table. Data can be moved using either an online method or an offline method. See the related links for information about converting existing indexes and about block indexes.

Related concepts:

"Table partitioning and multidimensional clustering tables" in Partitioning and Clustering Guide

"Block indexes for MDC tables" in Partitioning and Clustering Guide

"Indexes on partitioned tables" in Database Administration Concepts and Configuration Reference

Related tasks:

"Converting existing indexes to partitioned indexes" in Partitioning and Clustering Guide

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 12. 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.

Table 12. Summary of new Version 9.7 database manager configuration parameters (continued)

Parameter name	Description	Details
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_svcname	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, new ranges, or new values in Version 9.7.

Table 13. Summary of database manager configuration parameters with changed behaviors, new ranges, or new values

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.
diagpath	This parameter allows you to specify the fully qualified path for DB2 diagnostic information and split the diagnostic data into separate directories	The values "\$h", "pathname \$h", "\$n", "pathname \$n", "\$h\$n", and "pathname \$h\$n" are available in DB2 Version 9.7 Fix Pack 1 and later fix packs. You can specify to store DB2 diagnostic data in separate directories named according to the physical host, database partition, or both by setting the enhanced diagpath database manager configuration parameter.

Related concepts:

"Some database configuration parameters have been changed" on page 238

Related reference:

"RESET DATABASE CONFIGURATION " in Command Reference

"Configuration parameters summary" in Database Administration Concepts and Configuration Reference

"DB2 server behavior changes" in Upgrading to DB2 Version 9.7

FP1: Collection interval time for workload management statistics has changed

Starting with Version 9.7 Fix Pack 1, the workload management statistics collection interval is synchronized relative to a fixed start time (a day of the week and an hour of the day), rather than relative to when the DB2 database was activated.

Details

The fixed start time is Sunday at 00:00:00 (hh:mm:ss). If, for example, the catalog partition activation time is 22:30:00 on a Saturday night and the collection interval is set to 60 minutes, the first collection occurs 30 minutes later (conforming to the top-of-the-hour start time for the hour collection interval). However, if the catalog partition activation time is 22:30:00 on a Saturday night and the collection interval is set to 1 week (10,080 minutes), the first collection occurs 1 hour 30 minutes later (conforming to the day-of-the-week start time for the week collection interval).

Note: The collection interval is not compensated due to changes to the fixed start time that result from daylight savings time transitions. The interval, during which a transition from standard time to daylight savings time or vice versa occurs, could be shortened or lengthened compared to the setting of the `wlm_collect_int` database configuration parameter.

Resolution

No user response is required.

Related reference:

"wlm_collect_int - Workload management collection interval configuration parameter" in Workload Manager Guide and 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 14. 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.
DB2_SMS_TRUNC_TMPTABLE_THRESH	0	Starting in Version 9.7 Fix Pack 2, the default for this variable is -2, which means that there will not be any unnecessary file system access for any spilled SMS temporary objects whose size is less than or equal to 1 extent * number of containers. Temporary objects that are larger than this are truncated to 0 extent

Table 15. Registry variables with new values

Registry variable	New values
DB2AUTH	Starting in DB2 Version 9.7 Fix Pack 5, this variable has a new option, SQLADM_NO_RUNSTATS_REORG, which allows you to take away the ability of users with SQLADM authority from performing runstats or reorg operations.
DB2_EVMON_STMT_FILTER	Starting in DB2 Version 9.5 Fix Pack 1, this variable has new options that allow you to determine which rules to apply to which event monitors. Each option represents an integer value mapping to a specific SQL operation. .
DB2_FCM_SETTINGS	Starting in DB2 Version 9.7 Fix Pack 3, this variable has a new setting, FCM_CFG_BASE_AS_FLOOR, which allows you to set a floor value for the <i>fcm_num_buffers</i> and <i>fcm_num_channels</i> database manager configuration parameters so that auto-configuration will not tune below their set value.
DB2_SQLROUTINE_PREPOPTS	This variable has two new options: <ul style="list-style-type: none"> • 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
DB2_WORKLOAD	This variable has new values: CM, COGNOS_CS, FILENET_CM, MAXIMO, MDM, WAS, WC and WP. These settings allow you to configure a set of registry variables in your database for applications provided by IBM Content Manager, Cognos® Content Server, Filenet Content Manager, Maximo®, Master Data Management, WebSphere® Application Server, IBM WebSphere Commerce and WebSphere Portal. The CM and WC values are also available starting in DB2 Version 9.5 Fix Pack 3 and Fix Pack 4, respectively. The COGNOS_CS, FILENET_CM, MAXIMO, MDM, WAS, and WP values are also available starting in DB2 Version 9.5 Fix Pack 5.

Changed behaviors

Table 16. 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_ITP_LEVEL	This registry variable was introduced in Version 9.7 Fix Pack 2 to enable the parallel reading of data during certain types of backup operations. Starting in Version 9.7 Fix Pack 3, the settings of this registry variable are ignored and have no effect during backup operations.
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_RESTORE_GRANT_ADMIN_AUTHORITIES	Starting in Version 9.7 Fix Pack 5, if DB2_RESTORE_GRANT_ADMIN_AUTHORITIES is set to ON and you are restoring to an existing database, then SECADM and DBADM authorities are granted to the user that issues the restore operation.
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.
DB2_WORKLOAD	Starting in Version 9.7 Fix Pack 2, when this aggregate registry variable is set to SAP, the DB2_EXTENDED_OPTIMIZATION registry variable will be set to IXOR to improve performance for queries generated by SAP applications.

New variables

These environment and registry variables are new in Version 9.7:

Table 17. Environment and Registry variables added for Version 9.7

Registry variable	Description
DB2_ATS_ENABLE	This registry variable enables or disables the administrative task scheduler.

Table 17. Environment and Registry variables added for Version 9.7 (continued)

Registry variable	Description
DB2_BACKUP_USE_DIO	This registry variable allows you to have a backup image file directly written to disk, bypassing the file cache, potentially leading to better memory utilization on Linux platforms. This registry variable is available in Version 9.7 Fix Pack 6 and later fix packs.
DB2_DDL_SOFT_INVAL	This registry variable enables <i>soft invalidation</i> of applicable database objects when they are dropped or altered, meaning that active access to an object that is being invalidated can continue.
DB2_DEFERRED_PREPARE_SEMANTICS	This registry variable allows dynamic statements containing untyped parameter markers to use deferred prepare semantics. When set to YES, 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, or when set to NO (the default) compilation of such dynamic statements fails.
DB2_FCM_SETTINGS	On Linux operating systems, you can set this registry variable with the FCM_MAXIMIZE_SET_SIZE token to preallocate a default 4 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_HADR_ROS	This variable enables the HADR reads on standby feature. When DB2_HADR_ROS is enabled on the HADR standby database, the standby accepts client connections and allows read-only queries to run on it. This registry variable is available in Version 9.7 Fix Pack 1 and later fix packs.

Table 17. Environment and Registry variables added for Version 9.7 (continued)

Registry variable	Description
DB2_HISTORY_FILTER	This variable specifies operations that are not to modify the history file, thereby reducing potential contention on the history file. This registry variable is available in Version 9.7 Fix Pack 6 and later fix packs.
DB2_LIMIT_FENCED_GROUP	On Windows operating systems, if you enable extended security, this registry variable can limit the operating system permissions of the fenced routines that run under the fenced mode process (db2fmp).
DB2_NCHAR_SUPPORT	This variable allows the use of NCHAR, NVARCHAR, and NCLOB data types in Unicode databases. When it is set to ON, various national character related functions such as NCHAR() and TO_NCHAR() are also supported.
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.
DB2_PMODEL_SETTINGS	Starting in Version 9.7 Fix Pack 3, you can use this variable to modify the behavior of various aspects of the DB2 internal infrastructure. If you set this variable with the MLN_REMOTE_LISTENER option, applications can connect directly to each logical database partition instead of routing requests through the database partition server that is assigned to logical port 0. If you set this variable with the ENHANCED_ROLLBACK option, rollback requests for units of work are sent only to logical database partitions that participated in the transaction.

Table 17. Environment and Registry variables added for Version 9.7 (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.
DB2_SAS_SETTINGS	Starting in Version 9.7 Fix Pack 6, this DB2 registry variable is the primary point of configuration for in-database analytics with the SAS embedded process.
DB2_SQLWORKSPACE_CACHE	This variable allows you to control the amount of caching of previously used sections in the SQL Workspace. You would adjust the setting for DB2_SQLWORKSPACE_CACHE based on how much of the SQL workspace you want to make available for reuse, which can result in some performance improvements for OLTP workloads.
DB2_STANDBY_ISO	This variable coerces the isolation level requested by applications and statements running on an active HADR standby database to Uncommitted Read (UR). When DB2_STANDBY_ISO is set to UR, isolation levels higher than UR are coerced to UR with no warning returned. This registry variable is available in Version 9.7 Fix Pack 1 and later fix packs.
DB2STMM	Starting in DB2 Version 9.7 Fix Pack 5, you can use this registry variable to modify the self tuning memory manager (STMM) shared memory segment in order to mitigate downlevel instance compatibility issues.
DB2_USE_FAST_PREALLOCATION	This variable allows the fast allocation file system feature to reserve table space, and speed up the process of creating or altering large table spaces and database restore operations. This variable is available with DB2 Version 9.7 Fix Pack 1.

Related concepts:

"HADR reads on standby feature" in Data Recovery and High Availability Guide and Reference

"Some registry and environment variables have been deprecated" on page 295

"Some registry and environment variables have been discontinued" on page 312

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. Instead, the database manager can open individual log files 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 required 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.

AUTOCONFIGURE command has been changed

The values generated by the **AUTOCONFIGURE** command (and the Configuration Advisor) will not be the same as in previous releases because the usage of the **mem_percent** parameter has been changed.

Details

Starting in Version 9.7, the **mem_percent** parameter of the **AUTOCONFIGURE** command indicates the percentage of the **instance_memory** database manager configuration parameter instead of the total physical memory on the computer.

User response

If you do not specify a value for `mem_percent`, the percentage is calculated based on memory usage in the instance and the system up to a maximum of 25% of the instance memory.

Related reference:

"AUTOCONFIGURE " in Command Reference

"AUTOCONFIGURE command using the ADMIN_CMD procedure" in Administrative Routines and Views

The CONCURRENTDBCOORDACTIVITIES threshold has been changed

To reduce the chance of irreconcilable queue-based contention scenarios, the behavior of the CONCURRENTDBCOORDACTIVITIES threshold has been changed.

Details

When creating a CONCURRENTDBCOORDACTIVITIES threshold, irreconcilable queue-based contention scenarios occur 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 irreconcilable queue-based contention 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

Before using CONCURRENTDBCOORDACTIVITIES thresholds, be sure to become familiar with the effects that they can have on the database system. For more

information, see the "CONCURRENTDBCOORDACTIVITIES threshold" topic.

Related reference:

"CONCURRENTDBCOORDACTIVITIES threshold" in Workload Manager Guide and Reference

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

FP1: Detach operation for data partitions has been changed

In Version 9.7 Fix Pack 1 and later fix packs, the process to detach a data partition from a partitioned table is a two-phase process.

Details

When you issue the **ALTER TABLE** statement with the **DETACH** partition clause, the data partition that you are detaching is converted into a stand-alone table in the following two-phase process:

1. The **ALTER TABLE** operation logically detaches the data partition from the partitioned table. The data partition name is changed to a system-generated name of the form `SQLLyymmddhhmmssxxx` so that a subsequent attach can reuse the detached partition name immediately. In `SYSCAT.DATAPARTITIONS`, the status of the partition is set to L (logically detached) if there are no detached dependent tables or D if there are detached dependent tables.
2. An asynchronous partition detach task converts the logically detached partition into a stand-alone table.

The target table is unavailable until the asynchronous partition detach task completes the detach. For example, a **DROP** statement that drops the target table after a detach must wait until the asynchronous partition detach task completes the detach. In Version 9.7 and earlier releases, the target table of an **ALTER TABLE** statement with the **DETACH PARTITION** clause became available immediately after the transaction issuing the **ALTER TABLE** statement committed if there were no detached dependent tables that needed to be incrementally maintained with respect to the detached data partition. If there were detached dependent tables, the target table became available after the **SET INTEGRITY** statement is run on all detached dependent tables.

User response

Because the data partition name is changed to a system-generated name during the first phase of the detach process, you might need to modify applications that query the catalog views for detached data partitions and use the data partition names.

Related concepts:

"Asynchronous partition detach for data partitioned tables" in Partitioning and Clustering Guide

"Data partition detach phases" in Partitioning and Clustering Guide

Related reference:

"DROP " in SQL Reference, Volume 2

FP1: XML schema maxOccurs attribute values greater than 5000 are parsed differently

Starting in Version 9.7 Fix Pack 1, if you specify a value greater than 5000 for the maxOccurs attribute for an element in an XML schema definition, the XML parser treats the value as if you specified "unbounded".

Details

A maxOccurs attribute value of unbounded indicates that the element can appear an unlimited number of times. In that case, starting in Fix Pack 1, an XML document might pass validation when you use the XMLVALIDATE function even if the number of occurrences of an element exceeds the maximum according to the XML schema that you used to validate the document.

User Response

If you use an XML schema that defines an element that has a maxOccurs attribute value that is greater than 5000 and you want to reject XML documents that have a maxOccurs attribute value greater than 5000, you can define a trigger or procedure to check for that condition. In the trigger or procedure, use an XPath expression to count the number of occurrences of the element and return an error if the number of elements exceeds the maxOccurs attribute value

For example, the following trigger ensures that a document never has more than 6500 phone elements:

```
CREATE TRIGGER CUST_INSERT
  AFTER INSERT ON CUSTOMER
  REFERENCING NEW AS NEWROW
  FOR EACH ROW MODE DB2SQL
  BEGIN ATOMIC
    SELECT CASE WHEN X <= 6500 THEN 'OK - Do Nothing'
              ELSE RAISE_ERROR('75000', 'TooManyPhones') END
  FROM (
    SELECT XMLCAST(XMLQUERY('$INFO/customerinfo/count(phone)') AS INTEGER) AS X
    FROM CUSTOMER
    WHERE CUSTOMER.CID = NEWROW.CID );
END
```

Related reference:

"XMLVALIDATE " in SQL Reference, Volume 1

"Restrictions on the pureXML feature" in pureXML Guide

FP5: History file no longer locked during automatic deletion of recovery objects

When you set the **auto_del_rec_obj** database configuration parameter to ON or you perform a prune operation that invokes the delete syntax, the database manager deletes the corresponding log files, backup images, and load copy images when pruning the history file. In Version 9.7 Fix Pack 5 and later fix packs, the history file is no longer exclusively locked during the deletions.

This change helps improve the performance of the database because history file access is blocked for only a very short period of time, so log files can still be created.

Details

In previous releases and fix packs, some transactions could have been halted or delayed so that unnecessary files were being deleted from the history file. In some cases, you had to work around this by running manual prune operations during low-impact periods or increase the log file size to reduce the number of files to clean up.

Resolution

No user response is required

FP5: Some administrative routines and views have changed

In Version 9.7, Fix Pack 5 and later fix packs, there have been additional return fields included in some administrative routines and views.

Several table functions and administrative views are enhanced in Version 9.7, Fix Pack 5. These monitoring routines now return additional information about your databases and corresponding systems.

MON_BP_UTILIZATION

Now returns information about:

- AVG_SYNC_READ_TIME
- AVG_ASYNC_READ_TIME
- AVG_SYNC_WRITE_TIME
- AVG_ASYNC_WRITE_TIME

MON_GET_BUFFERPOOL

Now returns information about:

- POOL_ASYNC_READ_TIME
- POOL_ASYNC_WRITE_TIME
- BP_CUR_BUFFSZ

MON_GET_TABLE

Now returns information about:

- DATA_OBJECT_PAGES
- LOB_OBJECT_PAGES

- LONG_OBJECT_PAGES
- INDEX_OBJECT_PAGES
- XDA_OBJECT_PAGES

MON_GET_TABLESPACE

Now returns information about:

- POOL_ASYNC_READ_TIME
- POOL_ASYNC_WRITE_TIME
- TBSP_TRACKMOD_STATE

Related concepts:

“FP5: Table space modification status can be checked” on page 193

“FP5: Some monitoring routines and views are deprecated” on page 302

Related reference:

“MON_GET_BUFFERPOOL table function - Get buffer pool metrics” in Administrative Routines and Views

“MON_GET_TABLESPACE table function - Get table space metrics” in Administrative Routines and Views

“MON_GET_TABLE table function - get table metrics” in Administrative Routines and Views

“MON_BP_UTILIZATION - Retrieve metrics for bufferpools” in Administrative Routines and Views

Database setup and product installation changes summary

Version 9.7 includes changed functionality that affects how you install and setup DB2 databases.

To take advantage of the new features included in the release, some of the minimum software requirements have been updated. To ensure that your systems are correctly setup, review the “Installation requirements for DB2 database products” and “Support for elements of the database application development environment” topics.

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 concepts:

"Upgrade essentials for clients" in Upgrading to DB2 Version 9.7

"Upgrade essentials for database applications " in Upgrading to DB2 Version 9.7

"Upgrade essentials for routines" in Upgrading to DB2 Version 9.7

"Upgrade essentials for DB2 servers" in Upgrading to DB2 Version 9.7

Related reference:

"Support for elements of the database application development environment" in Getting Started with Database Application Development

"Installation requirements for DB2 database products" in Installing DB2 Servers

Licensing control for DB2 Express, DB2 Workgroup Edition, and workload management has been changed

In Version 9.7, DB2 Express® and DB2 Workgroup Server Edition use a hard-stop license enforcement policy with regards to CPU and memory usage. In addition, workload management (WLM) functionality provided through the DB2 Performance Optimization Feature is only available if the DB2 Performance Optimization Feature license key has been registered.

Details

The DB2 database manager checks for additional licensing compliance in the following scenarios:

- If an attempt is made to use the DB2 WLM functionality provided in the DB2 Performance Optimization Feature without the DB2 Performance Optimization Feature license key being registered, the SQL8029N message is returned.
- The CPU and memory resources available to the DB2 Express and Workgroup Server Edition products are limited to the capacity specified by the license. You can use DB2 Express and DB2 Workgroup Server on a system that has more capacity, but will only be able to exploit the capacity specified by the license.

User response

- Purchase the license key for the DB2 Performance Optimization Feature to use WLM from your IBM representative or authorized dealer. You will then need to update your license using the License Center or the **db2licm** command line utility.
- To take full advantage of your server's memory and CPU capacity, contact your IBM representative or authorized dealer to obtain a DB2 product with a higher licensed limit.

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 18. 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.

Table 18. New Version 9.7 database configuration parameters (continued)

Parameter name	Description	Details
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, and lock-related messages notification level. During database upgrade, these parameters are set to NONE, except for mon_deadlock which is set to WITHOUT_HIST, mon_lw_thresh which is set to 5 000 000, mon_lck_msg_lvl which is set to 1, and mon_pkglist_sz which is set to 32, 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_lck_msg_lvl	Monitoring lock event notification messages	
mon_obj_metrics	Monitoring object metrics	
mon_pkglist_sz	Monitoring package list size	
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 19. 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 4 KB)

The following database configuration parameters have changed behaviors or have new ranges in Version 9.7.

Table 20. 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 <i>Troubleshooting and Tuning Database Performance</i> .

Table 20. Database configuration parameters with changed behaviors or new ranges (continued)

Parameter name	Description	Version 9.7 change
database_memory	Database shared memory size	The self tuning memory manager (STMM) has an improved ability to adjust the database shared memory usage on Solaris Operating Environment. If database_memory is set to AUTOMATIC on a Solaris operating system, the database manager uses pageable memory for the database shared memory. As a result, the DB2 database system uses smaller memory pages by default and you might notice some performance degradation.
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 <i>Database Administration Concepts and Configuration Reference</i> .
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 254. 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	
num_db_backups	Number of database backups	Starting in Fix Pack 5, DB2 counts merged backups as full, non-incremental backups when determining the number of backups to retain.
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 21. Summary of deprecated database configuration parameters

Parameter name	Description	Details and resolution
<code>dyn_query_mgmt</code>	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 222

Related reference:

"RESET DATABASE CONFIGURATION " in Command Reference

"Configuration parameters summary" in Database Administration Concepts and Configuration Reference

"DB2 server behavior changes" in Upgrading to DB2 Version 9.7

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:

"New response file keywords have been added" on page 185

"Some response file keywords have been deprecated" on page 300

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 outside the profile registries 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.

FP2: DB2 Advanced Copy Services (ACS) is not automatically included in a compact installation

In Version 9.7 Fix Pack 2 and later fix packs, DB2 ACS is no longer installed automatically during DB2 installations of type compact.

Details

Before Version 9.7 Fix Pack 2, DB2 ACS was always installed during all DB2 product installations. Starting in Version 9.7 Fix Pack 2, DB2 ACS is an optional component during some types of DB2 product installations, including unattended installations. DB2 ACS is installed by default during DB2 installations of type **custom** and **typical**, and when you use the **db2_install** command.

During unattended installations, you can use the **ACS** response file keyword to install or uninstall DB2 ACS.

User response

If you have already completed a compact installation and must now install DB2 ACS, use a custom response file installation, or click **Work with Existing** from the DB2 Setup wizard to install only DB2 ACS into the DB2 copy that you already installed.

Related tasks:

"Installing DB2 Advanced Copy Services (ACS)" in Data Recovery and High Availability Guide and Reference

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 98
"Authorities overview" in Database Security Guide

Related reference:

"DB2 server behavior changes" in Upgrading to DB2 Version 9.7

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 98

"Authorities overview" in Database Security Guide

Related reference:

"DB2 server behavior changes" in Upgrading to DB2 Version 9.7

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 98

"Authorities overview" in Database Security Guide

Related reference:

"DB2 server behavior changes" in Upgrading to DB2 Version 9.7

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_svcname** 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 22. Mapping of server-side SSL support parameters to new database manager configuration parameters

Version 9.5 <code>SSLconfig.ini</code> parameters	Version 9.7 database manager configuration parameters
<code>DB2_SSL_KEYSTORE_FILE</code>	<code>ssl_svr_keydb</code>
<code>DB2_SSL_KEYSTORE_PW</code>	<code>ssl_svr_stash</code>
<code>DB2_SSL_KEYSTORE_LABEL</code>	<code>ssl_svr_label</code>
<code>DB2_SSL_LISTENER</code>	<code>ssl_svcename</code>

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 23. Mapping of client-side SSL support parameters to new database manager configuration parameters

Version 9.5 <code>SSLClientconfig.ini</code> parameters	Version 9.7 database manager configuration parameters
<code>DB2_SSL_KEYSTORE_FILE</code>	<code>ssl_clnt_keydb</code>
<code>DB2_SSL_KEYRING_STASH_FILE</code>	<code>ssl_clnt_stash</code>

Related concepts:

"SSL client support expanded and configuration simplified" on page 101

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

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.

Related reference:

"AUDIT_ARCHIVE procedure and table function - Archive audit log file" in Administrative Routines and Views

"AUDIT_DELIM_EXTRACT - performs extract to delimited file" in Administrative Routines and Views

"AUDIT_LIST_LOGS table function - Lists archived audit log files" in Administrative Routines and Views

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 DATAACCESS 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 24. 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: <ul style="list-style-type: none"> 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

Table 24. Authorization changes for Net Search Extender commands (continued)

Command	Version 9.5 authorization	Version 9.7 authorization
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 DATAACCESS authorities, otherwise the DB2 Text Search commands and stored procedures will fail even if the user has the correct authorities and privileges. In the Windows environment, if the DB2 text search service is run by a local System, then the system and the local system needs to have DBADM along with DATAACCESS authorities for the DB2 text search commands to run.

In addition, the authorities and privileges required for running the DB2 Text Search commands and stored procedures have changed as follows:

Table 25. 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: <ul style="list-style-type: none"> 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

Table 25. Authorization changes for the db2ts command (continued)

db2ts command	Version 9.5 authorization	Version 9.7 authorization
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 26. 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: <ul style="list-style-type: none"> 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. Starting in Version 9.7 Fix Pack 1 and later fix packs, users with SECADM authority can grant the necessary DBADM with DATAACCESS privileges to the instance owner by specifying the **AUTOGRANT** option when running the **db2ts ENABLE DATABASE FOR TEXT** command.

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.

Default value or behavior of selected IBM Data Server Driver keywords have changed

In Version 9.7 Fix Pack 6 and later fix packs, the default values for the **QueryTimeoutInterval**, and **ClientWorkstationName** IBM Data Server Driver keywords have changed. Also, the behavior of **ConnectionTimeout** keyword has changed.

Details

The **QueryTimeoutInterval** keyword specifies the time interval in seconds before the query timeout is checked. The default value for the **QueryTimeoutInterval** keyword is now 1 second.

The **ClientWorkstationName** keyword specifies the client workstation name that is sent to a database. The default value for the **ClientWorkstationName** keyword is now the host name of the client.

The **ConnectionTimeout** keyword specifies the time in seconds to wait for a reply when trying to establish a connection to a server before terminating the attempt. The **ConnectionTimeout** keyword value is enforced for a connection regardless of a high availability solution being enabled. If a connection attempt fails within the time specified by the **ConnectionTimeout** value, control is returned to the application with an SQL30081N error.

Default values of selected automatic client reroute IBM Data Server Driver keywords have changed

Starting in Version 9.7 Fix Pack 6, the default values for the **maxRefreshInterval**, **maxACRRetries**, **MaxTransports**, **MaxTransportIdleTime**, and **MaxTransportWaitTime** IBM Data Server Driver keywords have changed.

Details

The **maxRefreshInterval** keyword specifies the maximum elapsed time in seconds before the server list is refreshed. The default value for the **maxRefreshInterval** keyword is now 10 seconds.

The **MaxTransports** keyword specifies the maximum number of connections that a requester can make to the data sharing group. The default value for the **MaxTransports** keyword for DB2 for z/OS is now 1000.

The **MaxTransportIdleTime** keyword specifies the maximum elapsed time in seconds before an idle transport is dropped. The default is now 60 seconds.

The **MaxTransportWaitTime** keyword specifies the number of seconds that the client waits for a transport to become available. The default is now 1 second.

Related concepts:

"Configuration of Sysplex workload balancing and automatic client reroute for non-Java clients" in Call Level Interface Guide and Reference, Volume 1

"Configuration of Informix high-availability support for non-Java clients" in Call Level Interface Guide and Reference, Volume 1

"Configuring client affinities in non-Java clients for connection to DB2 Database for Linux, UNIX, and Windows" in Call Level Interface Guide and Reference, Volume 1

"Configuring client affinities in non-Java clients for connection to Informix database server connections" in Call Level Interface Guide and Reference, Volume 1

"Configuration of DB2 Database for Linux, UNIX, and Windows workload balancing support for non-Java clients" in Call Level Interface Guide and Reference, Volume 1

Cursor stability (CS) level behavior for newly created databases has been changed

To reduce lock wait and deadlock scenarios when using the Cursor Stability (CS) isolation level, currently committed semantics have been introduced and is enabled by default when new databases are created. Where possible, a read operation returns the currently committed result, ignoring what might happen to an uncommitted operation.

Details

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 when currently committed semantics is enabled, a read operation does not necessarily wait for a change to a row to be committed before returning a value.

The new CS behavior is beneficial in high-throughput transaction-processing database environments. In such environments, waiting on locks is undesirable. This new behavior is also 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.

If you upgrade a database from a previous release, the new CS behavior is not automatically enabled. To use this feature on an upgraded database, you must enable it manually.

User response

You can disable currently committed semantics for new databases created under CS, or enable currently committed semantics for upgraded databases, using the following methods:

- At the database level, using the new **cur_commit** database configuration parameter
- At the application level (overriding the database setting), using the **CONCURRENTACCESSRESOLUTION** option of the **BIND** and **PRECOMPILE** command
- At the stored procedure level (overriding the database setting), using the **DB2_SQLROUTINE_PREOPTS** registry variable and the **SET_ROUTINE_OPTS** procedure

Related concepts:

“Cursor stability (CS) isolation level enhancements provide more concurrency” on page 70

“Currently committed semantics improve concurrency” in Troubleshooting and Tuning Database Performance

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 about 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
- SYSINSTALLOBJECTS (starting in Version 9.7 Fix Pack 1)
- 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 have been added:

- ADMIN_MOVE_TABLE
- ADMIN_MOVE_TABLE_UTIL

The following security scalar function has been added:

- AUTH_GET_INSTANCE_AUTHID

The following SQL procedures routine has 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_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
- MONREPORT.CONNECTION (starting in Version 9.7 Fix Pack 1)
- MONREPORT.CURRENTAPPS (starting in Version 9.7 Fix Pack 1)
- MONREPORT.CURRENTSQL (starting in Version 9.7 Fix Pack 1)
- MONREPORT.DBSUMMARY (starting in Version 9.7 Fix Pack 1)
- MONREPORT.LOCKWAIT (starting in Version 9.7 Fix Pack 1)
- MONREPORT.PKGCACHE (starting in Version 9.7 Fix Pack 1)
- 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.NOOB
- 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_APPL_LOCKWAITS (starting in Version 9.7 Fix Pack 1)
- MON_GET_BUFFERPOOL
- MON_GET_CONNECTION
- MON_GET_CONNECTION_DETAILS
- MON_GET_CONTAINER
- MON_GET_EXTENT_MOVEMENT_STATUS
- MON_GET_FCM
- MON_GET_FCM_CONNECTION_LIST (starting in Version 9.7 Fix Pack 2)
- MON_GET_INDEX
- MON_GET_LOCKS
- MON_GET_PKG_CACHE_STMT

- MON_GET_PKG_CACHE_STMT_DETAILS (starting in Version 9.7 Fix Pack 1)
- 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 monitor routines have been added. They produce row-based easy-to-read output from the XML documents returned by other routines or about the lock name:

- MON_FORMAT_LOCK_NAME (starting in Version 9.7 Fix Pack 1)
- MON_FORMAT_XML_COMPONENT_TIMES_BY_ROW (starting in Version 9.7 Fix Pack 1)
- MON_FORMAT_XML_METRICS_BY_ROW (starting in Version 9.7 Fix Pack 1)
- MON_FORMAT_XML_TIMES_BY_ROW (starting in Version 9.7 Fix Pack 1)
- MON_FORMAT_XML_WAIT_TIMES_BY_ROW (starting in Version 9.7 Fix Pack 1)

The following monitor views have been added:

- MON_BP_UTILIZATION (starting in Version 9.7 Fix Pack 1)
- MON_CONNECTION_SUMMARY (starting in Version 9.7 Fix Pack 1)
- MON_CURRENT_SQL (starting in Version 9.7 Fix Pack 1)
- MON_CURRENT_UOW (starting in Version 9.7 Fix Pack 1)
- MON_DB_SUMMARY (starting in Version 9.7 Fix Pack 1)
- MON_LOCKWAITS (starting in Version 9.7 Fix Pack 1)
- MON_PKG_CACHE_SUMMARY (starting in Version 9.7 Fix Pack 1)
- MON_SERVICE_SUBCLASS_SUMMARY (starting in Version 9.7 Fix Pack 1)
- MON_TBSP_UTILIZATION (starting in Version 9.7 Fix Pack 1)
- MON_WORKLOAD_SUMMARY (starting in Version 9.7 Fix Pack 1)

The following explain routines have been added:

- EXPLAIN_FROM_ACTIVITY (starting in Version 9.7 Fix Pack 1)
- EXPLAIN_FROM_CATALOG (starting in Version 9.7 Fix Pack 1)
- EXPLAIN_FROM_DATA (starting in Version 9.7 Fix Pack 1)
- EXPLAIN_FROM_SECTION (starting in Version 9.7 Fix Pack 1)

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 updated:

- WLM_GET_CONN_ENV (starting in Version 9.7 Fix Pack 2)
- WLM_GET_SERVICE_CLASS_AGENTS_V97 (starting in Version 9.7 Fix Pack 1)
- WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES_V97
- WLM_GET_SERVICE_SUBCLASS_STATS_V97 (starting in Version 9.7 Fix Pack 1)

- WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES_V97 (starting in Version 9.7 Fix Pack 1)
- WLM_GET_WORKLOAD_STATS_V97 (starting in Version 9.7 Fix Pack 1)
- WLM_SET_CONN_ENV (starting in Version 9.7 Fix Pack 2)

The following catalog view and stored procedure for the Spatial Extender and DB2 Geodetic Data Management Feature have been updated:

- DB2GSE.ST_GEOMETRY_COLUMNS (starting in Version 9.7 Fix Pack 5)
- ST_register_spatial_column (starting in Version 9.7 Fix Pack 5)

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_LOCK (deprecated starting in Version 9.7 Fix Pack 1)
- SNAP_GET_LOCKWAIT (deprecated starting in Version 9.7 Fix Pack 1)
- 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

The following administrative views have been deprecated in Version 9.7 Fix Pack 1:

- SNAPLOCK
- SNAPLOCKWAIT
- LOCKS_HELD
- LOCKWAITS

User response

To have access to new administrative routines in Version 9.7 Fix Pack 1 in databases created in Version 9.7 prior to Fix Pack 1, you must have already run the **db2updv97** command. If your database was created before Version 9.7, it is not necessary to run the **db2updv97** command (because the system catalog is automatically updated by the database upgrade).

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.

To minimize the impact of changes to system-defined routines and views, review .

Related reference:

"Upgrade impact from system catalog changes" in *Upgrading to DB2 Version 9.7*

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 116.

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(VALUE(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
```


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 +0.000000000000000E+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 quotation marks 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 sequence 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 27. Stored procedure values inserted to table t2

Column c1	DB2 Version 9.7 (passing by reference)	DB2 Version 9.5 (passing by value)
1	<d>1</d>	<d>1</d>
2	<d>1</d>	<d>1</d><d>1</d>
3	<d>1</d>	<d>1</d><d>1</d>
4	<a><d>1</d><c>2</c>	NULL
5	NODE ID preserved	NODE ID NOT preserved

Related concepts:

"Node identity" in pureXML Guide

Related reference:

"Axes" in XQuery Reference

"Node comparisons" in XQuery Reference

"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:

"IBM Data Server Driver 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

FP1: Some import and load file type modifiers can accept unspecified values in trailing fields

Starting in DB2 Version 9.7 Fix Pack 1, the file type modifiers DATEFORMAT, TIMEFORMAT, and TIMESTAMPFORMAT for the import and load utilities can

accept input that contains unspecified values in trailing fields. A default value will be used for all trailing fields with unspecified input values.

Details

For applications that use the **IMPORT** or **LOAD** commands, if you use the **DATEFORMAT**, **TIMEFORMAT**, or **TIMESTAMPFORMAT** modifiers to input data into **DATE**, **TIME**, or **TIMESTAMP** columns, you can input values with unspecified values for trailing fields. In this case, a default value will be used. The default is 1 for year, month, or day fields, and 0 for all other fields.

User response

Applications that load and import data can now accept input data that conforms to the new input specifications.

Example

Example 1: **timestampformat** = "YYYY-MM-DD HH:MM:SS"

- The input 2007-11-23 07:29: is accepted, and has the effective value of 2007-11-23 07:29:00.
- The input 2007-11-23 is accepted, and has the effective value of 2007-11-23 00:00:00.
- The input 2007-11-23 :29:00 is not accepted. Only trailing input can be unspecified.

Example 2: **dateformat** = "YYYYMMDD"

- The input 1999 is accepted, and has the effective value of 19990101.

Related reference:

"IMPORT " in Command Reference

"LOAD " in Command Reference

Routines with bulk collection operations need to be re-created

If you use bulk collection operations in your PL/SQL routines, then changes to bulk collection might require that you re-create your routines.

Details

PL/SQL routines containing bulk collection operations that were created in DB2 V9.7 FP3a must be re-created in order to run successfully in DB2 FP4 or later fix packs. Routines with bulk collection operations created in DB2 V9.7 FP3a that are not re-created will return errors when run in DB2 FP4 (SQLSTATE 58004) and DB2 FP5 (SQLSTATE 55023).

Resolution

In order to run the routine without errors, the routine must be re-created.

Related reference:

"BULK COLLECT INTO clause (PL/SQL)" in SQL Procedural Languages: Application Enablement and Support

Chapter 20. Deprecated functionality in Version 9.7

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 in Version 9.7:

- Control Center tools (see “Control Center tools have been deprecated” on page 276)
- DB2 administration server (see “DB2 administration server (DAS) has been deprecated” on page 286)
- DB2 Governor and Query Patroller (see “DB2 Governor and Query Patroller have been deprecated” on page 286)
- Health monitor (see “Health monitor has been deprecated” on page 288)
- DB2 Health Advisor (see “FP4: DB2 Health Advisor has been deprecated” on page 289)
- IBM DB2 Geodetic Data Management Feature (see “FP5: IBM DB2 Geodetic Data Management Feature software has been deprecated” on page 290)
- Visual Studio 2005 support (see “FP5: Microsoft Visual Studio 2005 support has been deprecated” on page 290)

Additionally, the following functionality about general DB2 support has been deprecated in Version 9.7:

- LONG VARCHAR and LONG VARGRAPHIC data types (see “LONG VARCHAR and LONG VARGRAPHIC data types have been deprecated” on page 291)
- Worksheet Format (WSF) for Export and Load utilities (see “Worksheet Format (WSF) for Import and Export utilities has been deprecated” on page 291)
- LIST TABLESPACES and LIST TABLESPACE CONTAINERS commands and related APIs (see “LIST TABLESPACES and LIST TABLESPACE CONTAINERS commands have been deprecated” on page 292)
- 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 292)
- sqlugrpn API (see “sqlugrpn API has been deprecated” on page 293)
- sqlugtpi API (see “sqlugtpi API has been deprecated” on page 293)
- DB2SE_USA_GEOCODER (see “DB2SE_USA_GEOCODER has been deprecated” on page 294)
- Subset of Net Search Extender features and commands (see “Subset of Net Search Extender features and commands have been deprecated” on page 294)
- Functionality related to the discontinued type-1 indexes (see “Type-1 indexes have been discontinued” on page 306)
- **DB2_CAPTURE_LOCKTIMEOUT** and the **DB2_SERVER_ENCALG** registry variables (see “Some registry and environment variables have been deprecated” on page 295)

- **-file** option of the **db2rfdpen** command (see “FP1: -file option of db2rfdpen command has been deprecated” on page 296)
- COBOL and FORTRAN language support for db2History APIs (see “FP2: COBOL and FORTRAN language support has been deprecated for db2History APIs” on page 297)
- HP-UX 32-bit client support (see “FP3: HP-UX 32-bit client support has been deprecated” on page 297)

The following monitoring functionality has been deprecated in Version 9.7:

- 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 298)
- CREATE EVENT MONITOR FOR TRANSACTIONS statement (see “CREATE EVENT MONITOR FOR TRANSACTIONS statement has been deprecated” on page 298)
- Some monitoring routines and views (see “FP5: Some monitoring routines and views are deprecated” on page 302)
- Reporting of metrics in details_xml by the statistics event monitor (see “FP6: Reporting of metrics in details_xml by the statistics event monitor has been deprecated” on page 303)

The following product installation and instance management functionality has been deprecated in Version 9.7:

- **-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 299)
- Instance and database migration commands and APIs (see “Instance and database migration commands and APIs have been deprecated” on page 299)
- MIGRATE_PRIOR_VERSIONS and CONFIG_ONLY response file keywords (see “Some response file keywords have been deprecated” on page 300)
- Distributed installation support with Microsoft Systems Management Server (see “FP4: Distributed installation support with Microsoft Systems Management Server is deprecated (Windows)” on page 301)

The following troubleshooting functionality has been deprecated in Version 9.7:

- The **-global** option for troubleshooting tools (see “FP4: The -global option for troubleshooting tools has been deprecated” on page 302)

Review each topic to find out more details and to plan for future changes. Review also Chapter 21, “Discontinued functionality in Version 9.7,” on page 305 that might affect your databases and existing applications.

Control Center tools have been deprecated

Starting in Version 9.7, the Control Center tools have been deprecated and might be removed in a future release.

Details

You can use instead IBM Data Studio and the IBM InfoSphere Optim tools for managing DB2 for Linux, UNIX, and Windows databases and developing data-centric applications. For more information about these recommended tools, see “Resolution” on page 278.

The following Control Center tools and related features have been deprecated:

- Activity Monitor
- Command Editor
- Configuration Assistant
- Control Center and associated wizards and advisors
 - Add Partitions launchpad
 - Alter Database Partition Group wizard
 - Backup wizard
 - Configuration advisor
 - Configure Database Logging wizard
 - Configure Multisite Update wizard
 - Create Cache Table wizard
 - Create Database wizard
 - Create Federated Objects wizard (Also known as Create Nicknames wizard)
 - Create Table Space wizard
 - Create Table wizard
 - Design advisor
 - Drop Partition launchpad
 - Health Alert Notification
 - Health Indicator Configuration launchpad
 - Load wizard
 - Recommendation advisor
 - Redistribute Data wizard
 - Restore wizard
 - Set Up Activity Monitor wizard
 - Set Up High Availability Disaster Recovery (HADR) Databases wizard
 - Storage Management Setup launchpad
 - Troubleshooting wizard
- Control Center plug-in extensions
- Event Analyzer
- Health Center
- Indoubt Transaction Monitor
- Journal
- License Center
- Memory Visualizer
- Query Patroller Center
- Satellite Administration Center
- Task Center
- User interface to access Spatial Extender functionality

- User interface to Visual Explain

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)
- **db2daslevel1** (Show DAS level command)
- **db2eva** (Event analyzer command)
- **db2hc** (Start Health Center command)
- **db2indbt** (Start Indoubt Transaction Monitor Center command)
- **db2journal1** (Start Journal command)
- **db2lc** (Start License Center command)
- **db2memvis** (Start Memory Visualizer center command)
- **db2tc** (Start Task Center command)

The Replication Center has not been deprecated. It is still available as part of the Replication tools.

Resolution

Start using the **AUTOCONFIGURE** command to get recommendations from the configuration advisor. Although the wizard interface for the configuration advisor is still supported, this wizard is deprecated and might be removed in a future release.

Start using the **db2adv** command to get recommendations from the design advisor. Although the wizard interface for the design advisor is still supported, this wizard is deprecated and might be removed in a future release. The design advisor is not deprecated.

The deprecated Control Center tools and related features are still supported in Version 9.7. Only Version 9.7 DB2 servers or earlier releases are supported. However, you should consider becoming familiar with the IBM Data Studio and IBM InfoSphere Optim tools and start using them in place of the Control Center tools. For more information, see Data management and application development tools.

IBM Data Studio and IBM InfoSphere Optim tools are available to perform similar tasks that you used to perform with the Control Center tools. These recommended tools provide enhanced features for those tasks.

The following table provides details about IBM tools included in DB2 database and IBM InfoSphere Warehouse product editions:

Table 28. IBM tool by IBM product edition

IBM tool	IBM product editions that include the tool
IBM Data Studio	All DB2 database products, excluding DB2 Express-C, and all IBM InfoSphere Warehouse product editions
IBM InfoSphere Optim Performance Manager Enterprise Edition	DB2 Advanced Enterprise Server Edition, DB2 Database Enterprise Developer Edition, IBM InfoSphere Warehouse Departmental Edition, and IBM InfoSphere Warehouse Advanced Departmental Edition
IBM InfoSphere Optim Performance Manager Extended Edition	IBM InfoSphere Warehouse Enterprise Edition and IBM InfoSphere Warehouse Advanced Enterprise Edition
IBM InfoSphere Optim Query Workload Tuner	DB2 Advanced Enterprise Server Edition and IBM InfoSphere Warehouse Advanced Enterprise Edition

The following table shows the recommended tools to use in place of the Control Center tools.

Table 29. Recommended tools versus Control Center tools

Control Center tool	IBM tool	Links to additional information
Activity Monitor Event Analyzer	IBM InfoSphere Optim Performance Manager	<p>Relevant information:</p> <ul style="list-style-type: none"> Monitoring with Optim Performance Manager at http://publib.boulder.ibm.com/infocenter/perfmgmt/v5r1/topic/com.ibm.datatools.perfmgmt.monitor.doc/p_monitor.html <p>Added benefits:</p> <ul style="list-style-type: none"> Monitoring profiles and system template descriptions at http://publib.boulder.ibm.com/infocenter/perfmgmt/v5r1/topic/com.ibm.datatools.perfmgmt.monitor.doc/sys_templates_monitor_profiles.html Reporting with Optim Performance Manager at http://publib.boulder.ibm.com/infocenter/perfmgmt/v5r1/topic/com.ibm.datatools.perfmgmt.monitor.doc/p_report.html End-to-end database monitoring for applications (Extended Insight Analysis dashboard) at http://publib.boulder.ibm.com/infocenter/perfmgmt/v5r1/topic/com.ibm.datatools.perfmgmt.ei.overview.doc/topics/ei_overview.html¹

Table 29. Recommended tools versus Control Center tools (continued)

Control Center tool	IBM tool	Links to additional information
Command Editor	IBM Data Studio ²	<p>Relevant information:</p> <ul style="list-style-type: none"> • Modifying SQL and XQuery statements in the SQL and XQuery editor at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.sqlxeditor.doc/topics/teditsql.html • DB2 command support at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.changeCmd.doc/topics/chxucextendedcmdsSup.html • Diagramming access plans with Visual Explain at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.visualexplain.data.doc/topics/tvetop.html <p>Added benefits:</p> <ul style="list-style-type: none"> • Using templates for SQL and XQuery at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.sqlxeditor.doc/topics/ctemplates.html • Validating SQL and XQuery statements at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.sqlxeditor.doc/topics/tvalidatingSQLstatements.html
Configuration assistant	IBM InfoSphere Optim Configuration Manager	<p>Relevant information:</p> <ul style="list-style-type: none"> • Exploring configuration information about data clients and data servers in your environment at http://publib.boulder.ibm.com/infocenter/cfgmgr/v2r1/topic/com.ibm.datatools.configmgr.mancfgs.doc/texploresystem.html <p>Added benefits:</p> <ul style="list-style-type: none"> • Exploring configuration changes at http://publib.boulder.ibm.com/infocenter/cfgmgr/v2r1/topic/com.ibm.datatools.configmgr.mancfgs.doc/texploresystem.html • Creating a client rule to control and optimize the configurations of the managed clients at http://publib.boulder.ibm.com/infocenter/cfgmgr/v2r1/topic/com.ibm.datatools.configmgr.mancfgs.doc/tcreatorule.html

Table 29. Recommended tools versus Control Center tools (continued)

Control Center tool	IBM tool	Links to additional information
Control Center	IBM Data Studio	<p>Relevant information:</p> <ul style="list-style-type: none"> • Connecting to databases at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.common.nav.doc/topics/cdbconnect_cont.html • Creating and altering data objects with the Data Object editor at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.schema.manager.server.extensions.doc/topics/ccreate_alter.html • Managing data at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.common.nav.doc/topics/cdata_cont.html • Analyzing impact and dependency for data objects at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.modeler.modelanalysis.doc/topics/timpact_analysis.html • Modifying privileges on data objects at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.database.accesscontrol.ui.doc/topics/tprivmod_dbaedit.html • Generating DDL scripts at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.fe.ui.doc/topics/cddl.html • Administering databases with task assistants at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.adm.doc/topics/chxutrundbadmcmds.html • Diagramming access plans with Visual Explain at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.visualexplain.data.doc/topics/tvetop.html • Updating and viewing database statistics at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.database.statistics.ui.doc/topics/cstats_cont.html • Modifying SQL and XQuery statements in the SQL and XQuery editor at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.sqlxeditor.doc/topics/teditsql.html • Connecting to a federated database at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.db2.luw.serverdiscovery.ui.doc/topics/iiymdconnectingfeddb.html • Changing federated objects at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.uom.ui.doc/topics/c_federatedobjects.html

Table 29. Recommended tools versus Control Center tools (continued)

Control Center tool	IBM tool	Links to additional information
Control Center	IBM Data Studio	<p>Added benefits:</p> <ul style="list-style-type: none"> • Making basic database object changes with a change plan at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.uom.ui.doc/topics/c_basic_luw_scenario.html • Change management by using forward engineering from a model or the compare and migrate wizard at http://www.ibm.com/developerworks/data/library/techarticle/dm-0904changemgmt/index.html • Copying database objects at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.om.ui.doc/topics/ccopydbobjects.html • Developing a Java application that uses pureQuery annotated methods at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.javatool.welcome.doc/topics/pdqwelcometop.html • Deploying routines and SQL scripts to multiple systems. See Deploying routines using deployment groups at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.deployment.manager.ui.doc/topics/c_deploy_mgr_rout.html
Control Center associated wizards and advisors	IBM Data Studio	<p>The Task assistants provide similar functionality to the Backup, Configure database logging, Create database, Load, Restore, and Set up HADR wizards in the Control Center tools. See Administering databases with task assistants at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.adm.doc/topics/chxutrundbadmcmds.html.</p> <p>Added benefits:</p> <ul style="list-style-type: none"> • Using links to find detailed information about the message number or SQL code in the Messages section at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.adm.doc/topics/chxucbadmtaskassist.html • Running commands on multiple objects at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.adm.doc/topics/chxucmultipleobjectsupport.html • Management of cluster members in DB2 pureScale environments at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.adm.doc/topics/chxucbadmoverview.html

Table 29. Recommended tools versus Control Center tools (continued)

Control Center tool	IBM tool	Links to additional information
Health Center IBM Data Studio Data Studio web console	IBM Data Studio web console IBM InfoSphere Optim Performance Manager	<p>Relevant information:</p> <ul style="list-style-type: none"> Monitoring database health and availability at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.db.web.health.doc/topics/monitoringdshm.html Monitoring status of utilities operating on your databases. See Use the Current health views at http://www.ibm.com/developerworks/data/tutorials/dm-1012datastudiohealth/section3.html Monitoring with Optim Performance Manager at http://publib.boulder.ibm.com/infocenter/perfmgmt/v5r1/topic/com.ibm.datatools.perfmgmt.monitor.doc/p_monitor.html <p>Added benefits:</p> <ul style="list-style-type: none"> Opening the Data Studio web console from the Data Studio full client at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.dsweb.eclipse.health.monitor.doc/topics/c_health_integration.html Accessing more advanced performance monitoring functionality in IBM InfoSphere Optim Performance Manager 4.1.0.1 or later. See Advanced performance monitoring in Optim Performance Manager (4.1.0.1) at http://www.ibm.com/developerworks/data/tutorials/dm-1012datastudiohealth/section7.html.
Memory Visualizer	IBM InfoSphere Optim Performance Manager	<p>Relevant information:</p> <ul style="list-style-type: none"> Monitoring with Optim Performance Manager (Memory dashboard) at http://publib.boulder.ibm.com/infocenter/perfmgmt/v5r1/topic/com.ibm.datatools.perfmgmt.monitor.doc/p_monitor.html <p>Added benefits in InfoSphere Optim tools:</p> <ul style="list-style-type: none"> Finding buffer pools with a low hit ratio and high activity at http://publib.boulder.ibm.com/infocenter/perfmgmt/v5r1/topic/com.ibm.datatools.perfmgmt.monitor.doc/bufferpool_scenario.html

Table 29. Recommended tools versus Control Center tools (continued)

Control Center tool	IBM tool	Links to additional information
Query Patroller Center	IBM InfoSphere Optim Performance Manager	<p>Relevant information:</p> <ul style="list-style-type: none"> Configuring and monitoring a workload management solution at http://publib.boulder.ibm.com/infocenter/perfmgmt/v5r1/topic/com.ibm.datatools.workload.doc/tparent_configurewlm.html Migrating from Query Patroller to DB2 workload manager at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.qb.upgrade.doc/doc/t0053461.html <p>Added benefits:</p> <ul style="list-style-type: none"> Reporting with Optim Performance Manager at http://publib.boulder.ibm.com/infocenter/perfmgmt/v5r1/topic/com.ibm.datatools.perfmgmt.monitor.doc/p_report.html Configuring autonomic performance objectives for workloads (IBM InfoSphere Optim Performance Manager Extended Edition) at http://publib.boulder.ibm.com/infocenter/perfmgmt/v5r1/topic/com.ibm.datatools.workload.doc/configuringapo.html Monitor and control active work with DB2 workload manager. See Frequently asked questions about DB2 workload manager at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.wlm.doc/doc/c0052604.html for differences between Query Patroller and DB2 workload manager.
Task Center	IBM Data Studio	<p>Relevant information:</p> <ul style="list-style-type: none"> Creating and managing jobs at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.db.web.jobmanager.doc/topics/job_create_and_manage.html Scheduling command scripts at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.adm.doc/topics/chxucfflinescripts.html <p>Added benefits:</p> <ul style="list-style-type: none"> Scheduling Executable/shell Scripts and configuring email notifications to report on job completion. For more information, see Managing jobs in IBM Data Studio at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.db.web.jobmanager.doc/topics/job_overview.html
User Interface to Spatial Extender	IBM Data Studio	<p>Access to Spatial Extender functionality from the Control Center is deprecated. Alternatively, you can use the InfoSphere Optim tools or DB2 CLP commands to perform similar tasks.</p> <p>Added benefits:</p> <ul style="list-style-type: none"> See Added benefits for the Control Center and Control Center associated wizards and advisors in this table.

Table 29. Recommended tools versus Control Center tools (continued)

Control Center tool	IBM tool	Links to additional information
Visual Explain	IBM Data Studio	<p>Use the SQL and XQuery editor, which provides access to Visual Explain and enhanced query tuning capabilities. See also Diagramming access plans with Visual Explain at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.visualexplain.data.doc/topics/tvetop.html.</p> <p>Added benefits:</p> <ul style="list-style-type: none"> • Creating and deploying optimization profiles for SQL statements at http://publib.boulder.ibm.com/infocenter/qrytuner/v3r1/topic/com.ibm.datatools.qrytune.sngqry.doc/topics/optprofiles.html • Use advisors to generate and act on recommendations for: <ul style="list-style-type: none"> – Collecting statistics for database objects using the Statistics Advisor at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.qrytune.sngqry.doc/topics/genrecsstats.html – Reviewing recommendations for improving SQL statement access paths using the Access Path Advisor at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.qrytune.sngqry.doc/topics/genrecsapa.html – Creating indexes that improve the performance of single SQL statements using the Index Advisor at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.qrytune.sngqry.doc/topics/genrecsindexes.html³ – Rewriting queries to resolve the problems that are identified by the Query Advisor at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.qrytune.sngqry.doc/topics/genrecsqa.html³
<p>Notes:</p> <ol style="list-style-type: none"> 1. is a separately priced feature for IBM InfoSphere Optim Performance Manager. It is part of IBM InfoSphere Optim Performance Manager Extended EditionIBM InfoSphere Optim Performance Manager Extended Edition. 2. IBM Data Studio consist of three components. You can install any combination of these components and DB2 database products into a common shared environment. 3. The Index Advisor and Query Advisor require an active license for or IBM InfoSphere Optim Query Workload Tuner. 		

Related concepts:

"Workload management roadmap" in Workload Manager Guide and Reference

Related information:

 [Migrating from DB2 Control Center to IBM Data Studio](#)

DB2 administration server (DAS) has been deprecated

The DB2 Administration Server (DAS) has been deprecated in Version 9.7 and might be removed in a future release.

Details

Also, the DAS is not supported in DB2 pureScale environments.

The DAS is only required for using the Control Center tools or performing remote administration. The Control Center tools have been deprecated.

Resolution

Start using the IBM Data Studio and IBM Optim tools in place of the Control Center tools. For details, see "Control Center tools have been deprecated" on page 276.

Start using software programs that use Secure Shell protocol (SSH) for remote administration. For example, you can configure the workbench in Data Studio to run SQL statements, utilities, and commands, or to browse and access files on remote servers using the Secure Shell (SSH) protocol.

Related information:

 [Configuration for remote operations](#)

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`
- `db2gov1g`

User response

You can manage workloads more effectively by using DB2 workload manager, which provides many more features.

Starting in Version 9.7 Fix Pack 1 and later fix packs, you can use a sample program (qpwlmmig.pl) that generates a script that will help migrate a Query Patroller environment to a WLM environment.

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 91

Related tasks:

"Migrating from DB2 Governor to DB2 workload manager" in Upgrading to DB2 Version 9.7

"Migrating from DB2 Query Patroller to DB2 workload manager using the sample script" in Workload Manager Guide and Reference

Health monitor has been deprecated

The health monitor and associated health indicators have been deprecated. You can use instead IBM Data Studio and IBM InfoSphere Optim tools for monitoring DB2 for Linux, UNIX, and Windows databases.

Details

The following related APIs, API options, and option 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 related 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 IBM Data Studio and IBM InfoSphere Optim tools. For a mapping between these recommended tools and the Control Center, see “Table of recommended tools versus Control Center tools” in *What’s New for DB2 Version 9.7*.

FP4: DB2 Health Advisor has been deprecated

Starting with Version 9.7 Fix Pack 4 and later fix packs, the DB2 Health Advisor has been deprecated and might be removed in a future release. The **db2has** command has also been deprecated.

Details

The DB2 Health Advisor Service at IBM provided analysis and a report containing the findings and recommendations concerning the health of your DB2 environment. This service was provided on a trial basis and it is no longer available.


Running the **db2has** command still initiates the collection of data but the DB2 Health Advisor Service is no longer available to provide a report.

If you have any questions or concerns, send an email to db2has@ca.ibm.com.

Resolution

Alternatively, you can use other tools to monitor the health of your DB2 environment such as, IBM Data Studio web console which is designed to monitor the health and availability of databases.

Related information:

 [Monitoring database health and availability](#)

 [Monitor DB2 for Linux, UNIX, and Windows databases with Data Studio Health Monitor](#)

FP5: IBM DB2 Geodetic Data Management Feature software has been deprecated

Starting with Version 9.7 Fix Pack 5, DB2 Geodetic Data Management Feature has been deprecated. Version 9.7 Fix Pack 6 or later fix packs in Fix Central for DB2 Spatial Extender do not contain the software for the DB2 Geodetic Data Management Feature.

Details

For new installations of this functionality or to apply the Version 9.7 Fix Pack 6 or later fix packs to existing installations, contact IBM support for details on how to obtain the software for this functionality.

In Version 9.7 Fix Pack 5 or earlier releases, the DB2 Geodetic Data Management Feature was a complement to DB2 Spatial Extender. The DB2 Geodetic Data Management Feature handled objects defined on the surface of the Earth in a continuous spherical fashion rather than a planar, x- and y-coordinate system.

Resolution

If you hold a license for the DB2 Geodetic Data Management Feature and you apply the Version 9.7 Fix Pack 6 or a later fix pack to an existing installation, follow the instructions provided by IBM support to download and install DB2 Geodetic Data Management Feature and feature license after applying the fix pack.

If you do not have a license for the DB2 Geodetic Data Management Feature in Version 9.7 or Version 9.5, contact your IBM sales representative to determine the best solution for your environment.

FP5: Microsoft Visual Studio 2005 support has been deprecated

Microsoft Visual Studio 2005 support has been deprecated and might be removed in a future release.

Details

Support for Visual Studio 2005 has been deprecated because most of the editions of Visual Studio 2005 have reached their end of support dates.

Visual Studio 2005 was first released in October 2005 and was later upgraded to support NET Framework 2.0. Support for .NET Framework 3.0 is offered via Windows Visual Studio 2005 extensions for .NET Framework 3.0.

To have access to support of .NET Framework 3.0 or later, you should use Visual Studio 2008 or later.

Resolution

Start using Visual Studio 2008 or Visual Studio 2010 before support for Visual Studio 2005 is discontinued. You can convert existing projects in Visual Studio 2005 to Visual Studio 2008 or Visual Studio 2010.

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 Import and Export 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.

Related reference:

"**MON_GET_TABLESPACE** table function - Get table space metrics" in Administrative Routines and Views

"**MON_GET_CONTAINER** table function - Get table space container metrics" in Administrative Routines and Views

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, re-create 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 tasks:

“Upgrading Java routines” in Upgrading to DB2 Version 9.7

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 293

Related reference:

"db2GetDistMap - Get distribution map" in Administrative API Reference

DB2SE_USA_GEOCODER has been deprecated

DB2SE_USA_GEOCODER has been deprecated and might be removed in a future release.

Details

The DB2SE_USA_GEOCODER is a sample geocoder that translates United States addresses into ST_Point data. It was installed as a component of the DB2 Spatial Extender and automatically registered. The geocoder is supported only in a limited number of operating systems. The reference data for this geocoder was developed in 2002 as a way to provide an example for customers and has not been updated. Providing an update to this reference data will result on licensing fees for customers that want to use it.

DB2 Spatial Extender supports vendor-supplied and user-supplied geocoders. With these geocoders, you are free to use different input or output than that of DB2SE_USA_GEOCODER. Also, you can choose a geocoder that is supported in the operating system of your choice.

Resolution

Start using vendor-supplied and user-supplied geocoders. For details about how to use them, see [How to use a geocoder or Integrating Custom Geocoders with the DB2 Spatial Extender](#).

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 key concepts" in Net Search Extender Administration and User's Guide

Some registry and environment variables have been deprecated

There are a number of registry variables that 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 30. 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 46.
DB2_QP_BYPASS_APPLICATIONS	The registry variable is being deprecated and might be removed in a future release because the DB2 Workload Manager provides a greatly enhanced set of workload management features that replaces the DB2 Query Patroller. For more information, see "DB2 Governor and Query Patroller have been deprecated" on page 286
DB2_QP_BYPASS_COST	The registry variable is being deprecated and might be removed in a future release because the DB2 Workload Manager provides a greatly enhanced set of workload management features that replaces the DB2 Query Patroller. For more information, see "DB2 Governor and Query Patroller have been deprecated" on page 286

Table 30. Registry and environment variables deprecated in Version 9.7 (continued)

Registry or environment variable	Details
DB2_QP_BYPASS_USERS	The registry variable is being deprecated and might be removed in a future release because the DB2 Workload Manager provides a greatly enhanced set of workload management features that replaces the DB2 Query Patroller. For more information, see “DB2 Governor and Query Patroller have been deprecated” on page 286
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 100.
DB2_USE_DB2JCCT2_JROUTINE	This variable is used for switching back to JDBC type 2 driver, which was deprecated in a previous release. This registry variable is being deprecated and might be removed in a future release when the JDBC type 2 driver is removed

Related concepts:

“Some registry and environment variables have changed” on page 225

FP1: -file option of db2rfpen command has been deprecated

Starting in Version 9.7 Fix Pack 1, the **-file** option of the reset rollforward pending state (**db2rfpen**) command has been deprecated and might be removed in a future release.

Details

When you use the **-file** option, only the specified log control file (SQLOGCTL.LFH.1 or SQLOGCTL.LFH.2) is updated. This causes the files to no longer be synchronized. As a result, if the database is placed into rollforward pending state using the primary log control file (SQLOGCTL.LFH.1) and this file later becomes unavailable, the database will no longer be in rollforward pending state. Similarly, if the database is placed into rollforward pending state using the secondary log control file (SQLOGCTL.LFH.2) and the primary log control file remains available, the database will not be in rollforward pending state.

User response

Use the **database_alias** parameter or the **-path** option instead.

Related reference:

"db2rfdpen - Reset rollforward pending state " in Command Reference

FP2: COBOL and FORTRAN language support has been deprecated for db2History APIs

Starting with DB2 Version 9.7 Fix Pack 2, COBOL and FORTRAN language support for the db2History APIs has been deprecated and might be discontinued in a future release.

Details

The following APIs are affected by the deprecated support of the COBOL and FORTRAN application development languages:

- db2HistoryCloseScan
- db2HistoryGetEntry
- db2HistoryOpenScan
- db2HistoryUpdate

User response

Use one of the following supported alternatives:

- Access history information through the DB_HISTORY administrative view.
- Use a different programming language to develop your administrative application.

Related reference:

"db2HistoryUpdate - Update a database history records entry" in Administrative API Reference

"db2HistoryOpenScan - Start a database history records scan" in Administrative API Reference

"db2HistoryGetEntry - Get the next entry in the database history records" in Administrative API Reference

"db2HistoryCloseScan - End the database history records scan" in Administrative API Reference

"db2HistoryData " in Administrative API Reference

"DB_HISTORY administrative view - Retrieve history file information" in Administrative Routines and Views

FP3: HP-UX 32-bit client support has been deprecated

Starting with DB2 Version 9.7 Fix Pack 3 , HP-UX 32-bit client support has been deprecated and might be discontinued in a future release.

Details

When support for HP Itanium-based servers was introduced in DB2 Version 8, 32-bit DB2 client libraries on HP-UX were provided to support customers and partners who could not immediately enable their 32-bit HP-UX PA RISC applications on native 64-bit Itanium environments. Support for DB2 server instances on HP-UX PA RISC was removed in DB2 Version 9.5. As HP Itanium-based servers are now common, 32-bit DB2 on HP-UX client support is deprecated and might be removed in future releases. This change does not affect

other platforms, where 32-bit DB2 client support continues to be enhanced and supported.

Resolution

32-bit applications on HP-UX support has been deprecated starting with Version 9.7 Fix Pack 3. Such support might be discontinued in future releases of DB2 and DB2 Connect releases. To prepare for the change, it is recommended to migrate 32-bit HP-UX application to 64-bit so that applications can run on native 64-bit HP-UX Itanium-based environments.

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 38

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

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 **sqlmgdb** and **sqlgmigdb** 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 31. 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.
sqlmgmdb 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	<p>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.</p> <p>Any existing response files should be changed to replace the deprecated keyword with the new keyword.</p>
CONFIG_ONLY	<p>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 db2i setup command. The keyword had two options: YES or NO.</p> <p>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 db2i setup, the DB2 code assumes the option to be YES indicating that the response file is performing configuration tasks only.</p> <p>Existing scripts or response files do not need to be modified. Even if this keyword exists in a response file of db2i setup, the keyword value is assumed to be YES, regardless of the value specified.</p>

Related concepts:

"New response file keywords have been added" on page 185

"INTERACTIVE response file keyword has been changed" on page 241

Related reference:

"Response file keywords" in Installing DB2 Servers

FP4: Distributed installation support with Microsoft Systems Management Server is deprecated (Windows)

Support to install DB2 products across a network and set up the installation from a central location using Microsoft Systems Management Server (SMS) has been deprecated.

Details

Starting with Version 9.7 Fix Pack 4, you can perform distributed installations of DB2 products using the Microsoft Systems Center Configuration Manager (SCCM).

Although you can still perform distributed installations of DB2 products using SMS, you should not use SMS for DB2 product installations as Microsoft has ended its support for this product.

Resolution

Use SCCM to perform distributed installations instead. For details, see the SCCM documentation at <http://technet.microsoft.com/en-us/library/cc507089.aspx>.

FP4: The -global option for troubleshooting tools has been deprecated

In DB2 Version 9.7 Fix Pack 4 and later fix packs, the -global option that is supported by a number of DB2 troubleshooting tools has been deprecated and might be removed in a future release.

Details

Previously, the -global option was used to collect diagnostic information about remote hosts and partitions. The functionality provided by the -global option has been deprecated and replaced with the -member and -host options for the following troubleshooting tools:

- **db2trc**
- **db2pd**
- **db2fodc**
- **db2pdcfg**
- **db2support**

Collecting diagnostic information globally previously required the -global option to be specified, even when remote hosts and partitions were also specified. To simplify the syntax, the mandatory use of the -global option on remote hosts and partitions is no longer required.

User response

If you currently use the -global option with the **db2trc**, **db2pd**, **db2fodc**, **db2pdcfg** or **db2support** command, start using the -member option or the -host option instead. The -member option can be used to specify any database partition number, while -host is used to specify any host. If you want to collect diagnostic information about all members globally, without having to specify every member in the system, you can use the -member all option.

Related concepts:

“FP4: The serviceability of large database systems has improved” on page 196

FP5: Some monitoring routines and views are deprecated

Starting in Version 9.7 Fix Pack 5, three monitoring table functions and their corresponding administrative views have been deprecated and might be removed in a future release.

Details

With the introduction of the two new table functions MON_GET_MEMORY_SET and MON_GET_MEMORY_POOL in Version 9.7 Fix Pack 5, the following table functions are deprecated:

- The SNAP_GET_DBM_MEMORY_POOL table function and SNAPDBM_MEMORY_POOL administrative view
- The SNAP_GET_DB_MEMORY_POOL table function and SNAPDB_MEMORY_POOL administrative view
- The SNAP_GET_AGENT_MEMORY_POOL table function and SNAPAGENT_MEMORY_POOL administrative view

In addition, the MON_GET_TABLE table function has been enhanced to include additional information about the number of disk pages used for storing different types of data objects. Because of this enhancement, the SNAP_GET_TAB_V91 table function and the SNAPTAB administrative view have been deprecated.

Resolution

Start using the new table functions MON_GET_MEMORY_SET and MON_GET_MEMORY_POOL to get information about memory usage, and the MON_GET_TABLE table function to obtain information about disk pages before the deprecated views and table functions are removed.

Related concepts:

"FP5: Some administrative routines and views have changed" on page 235

Related reference:

"MON_GET_TABLE table function - get table metrics" in Administrative Routines and Views

"MON_GET_MEMORY_SET - get memory set information " in Administrative Routines and Views

"MON_GET_MEMORY_POOL - get memory pool information" in Administrative Routines and Views

FP6: Reporting of metrics in details_xml by the statistics event monitor has been deprecated

Starting with Version 9.7 Fix Pack 6, the XML document details_xml, which is included in two of logical data groups for the statistics event monitor has been deprecated. This document might be removed from the output of this event monitor in a future release.

Details

When the logical data groups event_sstats and event_wlstats are included in the output of the statistics event monitor, the XML document details_xml, which contains monitor elements that report metrics information is included in the output of the event monitor. The elements that appear in this document are included in the XML document metrics, which is now included in these two logical data groups. In addition, two new logical data groups, event_scmetrics and event_wlmetrics have been added to the event monitor. These logical data groups include the metrics reported in the metrics document. See "FP6: XML document metrics stores system metrics collected by statistics event monitor" on page 56 for more information.

Resolution

If you use the XML metrics data returned in the details_xml document, start using the metrics document instead. Alternatively, if you include the logical data groups event_scmetrics and event_wlmetrics in the information collected by the event monitor, you can access the metrics monitor elements directly. For example, if you create a statistics event monitor that writes to tables that includes these two logical data groups, you can access the metrics monitor elements from the new tables associated with each of these groups. See "FP6: New logical data groups added to the statistics event monitor" on page 55 for more information.

Chapter 21. Discontinued functionality in Version 9.7

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 in Version 9.7:

- Type-1 indexes (see “Type-1 indexes have been discontinued” on page 306)
- Windows 32-bit partitioned databases (see “32-bit partitioned databases are no longer supported (Windows)” on page 307)
- Netscape browser support (see “Netscape browser support has been discontinued” on page 307)
- Support on some Linux distributions and DRDA connectivity to some DB2 database servers (see “Some operating systems are no longer supported”)

The following products are no longer supported in Version 9.7:

- XML Extender (see “XML Extender has been discontinued” on page 308)
- Web Object Runtime Framework (see “Web Object Runtime Framework (WORF) support has been discontinued” on page 308)
- DB2 embedded application server (see “DB2 embedded application server (EAS) has been discontinued” on page 309)
- DB2 Everyplace (see “IBM DB2 Everyplace products have been discontinued” on page 309)

The following APIs, commands, command options, and registry variable are discontinued in Version 9.7:

- **db2uidd1** command (see “db2uidd1 command has been discontinued” on page 309)
- **db2secv82** command (see “db2secv82 command has been discontinued” on page 310)
- **GET AUTHORIZATIONS** command (see “GET AUTHORIZATIONS command has been discontinued” on page 310)
- **sqladau** API (see “sqladau API and sql_authorization data structure have been discontinued” on page 311)
- **-a** and **-p** options of the **db2ilist** command (see “db2ilist command options -a and -p have been discontinued” on page 311)
- **DB2_THREAD_SUSPENSION** registry variable (see “Some registry and environment variables have been discontinued” on page 312)

Some operating systems are no longer supported

Starting in Version 9.7, support for some Linux distributions has been discontinued. Also, support has been discontinued for connecting to older releases of some DB2 products.

Details

The following Linux distributions are no longer supported:

- Red Hat Enterprise Linux (RHEL) 4
- SUSE Linux Enterprise Server (SLES) 9

- Ubuntu 7.x

In addition, DRDA connectivity to the following DB2 for z/OS and IBM DB2 for IBM i data servers is no longer supported:

- DB2 for z/OS, Version 7.1
- DB2 for i, V5R1
- DB2 for i, V5R2

User Response

Review the list of supported Linux distributions and the list of supported servers through DRDA connectivity.

Related reference:

"IBM i and mainframe support for DB2 Connect" in *Installing and Configuring DB2 Connect Servers*

"Installation requirements for DB2 database products" in *Installing DB2 Servers*

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 tasks:

"Converting type-1 indexes to type-2 indexes" in Upgrading to DB2 Version 9.7

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.

Related concepts:

"Deprecated or discontinued functionality that affects DB2 server upgrades" in Upgrading to DB2 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.

Related tasks:

"Migrating from XML Extender to pureXML" in Upgrading to DB2 Version 9.7

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 308.

If you use DB2 samples that require an application server, you can use the WebSphere Application Server Community Edition (CE) application server.

IBM DB2 Everyplace products have been discontinued

IBM DB2 Everyplace® products have been discontinued.

Details

IBM has withdrawn from marketing the IBM DB2 Everyplace products. It has also announced April 30, 2013 as the end of support date. See the announcement letter number ENUS911-026.PDF for details.

IBM solidDB is now the strategic relational in-memory database technology solution for delivering data with extreme speed and extreme availability. See the solidDB product Web page for details at <http://www.ibm.com/software/data/soliddb/soliddb/>.

Resolution

Use IBM solidDB products, or one of the open source relational databases such as Apache Derby or SQLite for delivering data with extreme speed.

Related information:

 [Apache Derby](#)

 [SQLite](#)

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 tasks:

"Converting type-1 indexes to type-2 indexes" in Upgrading to DB2 Version 9.7

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.

Related reference:

"AUTH_LIST_AUTHORITIES_FOR_AUTHID " in Administrative Routines and Views

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

"AUTH_LIST_AUTHORITIES_FOR_AUTHID " in Administrative Routines and Views

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 **db2is** command on the server and then the **db2ilist** command in each of the installation directories shown in the **db2is** 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 32. 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 58

Related concepts:

"Some registry and environment variables have been deprecated" on page 295

"Some registry and environment variables have changed" on page 225

Chapter 22. Summary of deprecated and discontinued DB2 functionality in Version 9.7 and earlier releases

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 314
- “Functionality deprecated in Version 9.1 and possibly discontinued in a later release” on page 316
- “Functionality deprecated in Version 9.5 and possibly discontinued in a later release” on page 319
- “Functionality deprecated in Version 9.7” on page 321

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 33. V9 discontinued functionality

Release	Links to additional information
Version 9.1	<ul style="list-style-type: none">• “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	<ul style="list-style-type: none">• “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	<ul style="list-style-type: none">• “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 34. 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	<ul style="list-style-type: none"> Alternate FixPak images are discontinued (UNIX)
Audio, Image, and Video (AIV) Extenders	Version 9.1	<ul style="list-style-type: none"> Audio, Image, and Video (AIV) Extenders are no longer supported
Autoloader utility (db2atld) and related registry variable	Version 9.1	<ul style="list-style-type: none"> Autoloader utility (db2atld) is no longer supported
buffpage database configuration parameter	To be determined	<ul style="list-style-type: none"> Version 8 incompatibilities with previous releases
CALL_RESOLUTION DEFERRED statement and sqleproc API	To be determined	<ul style="list-style-type: none"> Version 8 incompatibilities with previous releases
COLNAMES column in SYSCAT.INDEXES	To be determined	<ul style="list-style-type: none"> DB2 Universal Database™ planned incompatibilities
db2profc command	Version 9.1	<ul style="list-style-type: none"> db2sqljcustomize - DB2 SQLJ Profile Customizer Command db2profc and db2profp utilities are discontinued
Data Links Manager, related registry variables and configuration parameters	Version 9.1	<ul style="list-style-type: none"> Data Links Manager is no longer supported
Data Warehouse Center and Information Catalog Center	Version 9.1	<ul style="list-style-type: none"> The Data Warehouse Center and the Information Catalog Center are no longer included
DB2 Administration Tools on some platforms	Version 9.1	<ul style="list-style-type: none"> DB2 Administration Tools are no longer supported on some platforms
db2profp command	Version 9.1	<ul style="list-style-type: none"> db2sqljprint - DB2 SQLJ Profile Printer Command db2profc and db2profp utilities are discontinued
db2reg2large utility for converting DMS table space size	Version 9.1	<ul style="list-style-type: none"> db2reg2large utility for converting DMS table space size is discontinued
Desktop icon and folder-making utilities (Linux operating systems)	Version 9.1	<ul style="list-style-type: none"> Desktop icon and folder making utilities are no longer supported (Linux)
Extended Storage option for buffer pools	Version 9.1	<ul style="list-style-type: none"> Extended Storage option for buffer pools is discontinued
indexsort database configuration parameter	To be determined	<ul style="list-style-type: none"> Version 8 incompatibilities with previous releases
JDBC type 2 driver	To be determined	<ul style="list-style-type: none"> Introduction to Java application support
JDBC type 3 driver and related registry variable	Version 9.1	<ul style="list-style-type: none"> Java Database Connectivity (JDBC) Type 3 JDBC support is discontinued

Table 34. Functionality deprecated in Version 8 and possibly discontinued in a later release (continued)

Functionality	Discontinued in release	Links to additional information
NetBIOS and SNA communication protocols and related registry variables and configuration parameters	Version 9.1	<ul style="list-style-type: none"> NetBIOS and SNA communication protocols are no longer supported
Network monitor elements: <ul style="list-style-type: none"> max_network_time_2_ms max_network_time_8_ms max_network_time_32_ms max_network_time_gt32_ms 	To be determined	<ul style="list-style-type: none"> Some network time monitor elements have been deprecated
PK_COLNAMES and FK_COLNAMES in SYSCAT.REFERENCES	To be determined	<ul style="list-style-type: none"> DB2 Universal Database planned incompatibilities
Subset of ODBC 3.0 functions that affect CLI support	To be determined	<ul style="list-style-type: none"> For the list of deprecated features and additional information, see CLI and ODBC function summary
Text Extender	Version 9.1	<ul style="list-style-type: none"> Text Extender is no longer supported
Uncataloged stored procedures	To be determined	<ul style="list-style-type: none"> Version 8 incompatibilities with previous releases
Variable argument list stored procedures	To be determined	<ul style="list-style-type: none"> Version 8 incompatibilities with previous releases
Vendor escape clauses in CLI statements	To be determined	<ul style="list-style-type: none"> Vendor escape clauses in CLI applications
Vendor load API (sqluvtd)	Version 9.1	<ul style="list-style-type: none"> Vendor load API (sqluvtd) is discontinued
Virtual interface (VI) architecture usage in fast communications manager (FCM) and related registry variables	Version 9.1	<ul style="list-style-type: none"> FCM no longer uses VI architecture
VSE and VM objects in the DB2 Control Center	Version 9.1	<ul style="list-style-type: none"> VSE and VM objects are no longer supported in the DB2 Control Center

Table 35. 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	<ul style="list-style-type: none"> Deprecated and discontinued features
DB2JVVIEW	Version 9.1	<ul style="list-style-type: none"> Deprecated and discontinued features
DB2_LGPAGE_BP	Version 9.5	<ul style="list-style-type: none"> List of Version 8 known limitations, problems, and workarounds Some registry and environment variables are discontinued
DB2NOLIOAIO	Version 9.5	<ul style="list-style-type: none"> DB2NOLIOAIO registry variable replaced by DB2LINUXAIO (Linux) Some registry and environment variables are discontinued
DB2_SCATTERED_IO (Linux)	Version 9.1	<ul style="list-style-type: none"> DB2_SCATTERED_IO registry variable is discontinued (Linux)

Functionality deprecated in Version 9.1 and possibly discontinued in a later release

Table 36. 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> -w option for db2icrt, db2ilist, and db2iupdt is discontinued (Linux and UNIX)
CLISchema CLI keyword	Version 9.5	<ul style="list-style-type: none"> CLISchema CLI keyword is discontinued
COLNAMES column in SYSCAT.INDEXES	To be determined	<ul style="list-style-type: none"> COLNAMES column in SYSCAT.INDEXES is deprecated
Database logging using raw devices	To be determined	<ul style="list-style-type: none"> Database logging using raw devices is deprecated
db2ilist command options -a and -p (Linux and UNIX operating systems)	Version 9.7	<ul style="list-style-type: none"> db2ilist command options are deprecated (Linux and UNIX) db2ilist command options -a and -p have been discontinued
db2licm command option -n	Version 9.5	<ul style="list-style-type: none"> -n option of the db2licm command is discontinued
db2undgp command	Version 9.5	<ul style="list-style-type: none"> Deprecated and discontinued features db2undgp command is discontinued
db2samp1 command option -schema	Version 9.1, Fix Pack 2	<ul style="list-style-type: none"> The -schema option of the db2samp1 command is discontinued
db2secv82 command	Version 9.7	<ul style="list-style-type: none"> db2secv82 command is deprecated db2secv82 command has been discontinued
Default function entry points support in external routine libraries	To be determined	<ul style="list-style-type: none"> External routines now require an explicit entry point specification
DB2 Web Tools	Version 9.5	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> Extended storage (ESTORE) feature support is discontinued
iCheckPending parameter	To be determined	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> Network Information Services (NIS and NIS+) support is deprecated (Linux and UNIX)

Table 36. 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	<ul style="list-style-type: none"> • Deprecated and discontinued features • Some database manager configuration parameters have changed
Subset of SQL administrative routines	To be determined	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Type-1 indexes are deprecated • Type-1 indexes have been discontinued • db2uiddl command has been discontinued

Table 37. 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	<ul style="list-style-type: none"> • Deprecated and discontinued features • Some registry and environment variables are discontinued
DB2CCMSRV	Version 9.5	<ul style="list-style-type: none"> • Deprecated and discontinued features • Some registry and environment variables are discontinued
DB2_COMMIT_ON_EXIT	To be determined	<ul style="list-style-type: none"> • Deprecated and discontinued features
DB2_CORRELATED_PREDICATES	To be determined	<ul style="list-style-type: none"> • Deprecated and discontinued features
DB2_ENABLE_BUFPD	To be determined	<ul style="list-style-type: none"> • Deprecated and discontinued features
DB2LINUXAIO	Version 9.5	<ul style="list-style-type: none"> • DB2LINUXAIO registry variable is deprecated (Linux) • Some registry and environment variables are discontinued
DB2_MAPPED_BASE	To be determined	<ul style="list-style-type: none"> • Deprecated and discontinued features
DB2_NO_MPFA_FOR_NEW_DB	To be determined	<ul style="list-style-type: none"> • Deprecated and discontinued features
DB2_PRED_FACTORIZE	To be determined	<ul style="list-style-type: none"> • Deprecated and discontinued features
OS/2 related registry variables: <ul style="list-style-type: none"> • DB2UPMPR • DB2UPMSINGLE 	Version 9.5	<ul style="list-style-type: none"> • Deprecated and discontinued features • Some registry and environment variables are discontinued

Table 37. 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: <ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Deprecated and discontinued features
DB2_BLOCK_ON_LOG_DISK_FULL	Version 9.5	<ul style="list-style-type: none"> • Deprecated and discontinued features • Some registry and environment variables are discontinued
DB2_FORCE_FCM_BP	Version 9.5	<ul style="list-style-type: none"> • Deprecated and discontinued features • Some registry and environment variables are discontinued
DB2_LGPAGE_BP	Version 9.5	<ul style="list-style-type: none"> • Deprecated and discontinued features • Some registry and environment variables are discontinued
DB2_MEMALLOCATE_HIGH	Version 9.5	<ul style="list-style-type: none"> • Deprecated and discontinued features • Some registry and environment variables are discontinued
DB2_MIGRATE_TS_INFO	Version 9.5	<ul style="list-style-type: none"> • Deprecated and discontinued features • Some registry and environment variables are discontinued

Table 37. 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	<ul style="list-style-type: none"> • Deprecated and discontinued features • Some registry and environment variables are discontinued
DB2_NR_CONFIG	Version 9.5	<ul style="list-style-type: none"> • Deprecated and discontinued features • Some registry and environment variables are discontinued
DB2_OLAP_BUFFER_SIZE	Version 9.5	<ul style="list-style-type: none"> • 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 38. 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	<ul style="list-style-type: none"> • Some database manager configuration parameters have changed
app_ct1_heap_sz , appgroup_mem_sz , and groupheap_ratio configuration parameters	Version 9.5	<ul style="list-style-type: none"> • Some database configuration parameters have changed
DB2 embedded application server (EAS)	Version 9.7	<ul style="list-style-type: none"> • DB2 embedded application server (EAS) has been discontinued
GET AUTHORIZATIONS command	Version 9.7	<ul style="list-style-type: none"> • GET AUTHORIZATIONS command is deprecated • GET AUTHORIZATIONS command has been discontinued
IMPORT command options CREATE and REPLACE_CREATE	To be determined	<ul style="list-style-type: none"> • IMPORT command options CREATE and REPLACE_CREATE are deprecated
logretain and userexit configuration parameter	Version 9.5	<ul style="list-style-type: none"> • Some database configuration parameters have changed
Log control file SQLLOGCTL.LFH and the -file option of the db2f1sn command	SQLLOGCTL.LFH: Version 9.5 The -file option: To be determined	<ul style="list-style-type: none"> • Log control file SQLLOGCTL.LFH has been renamed and copied
LONG VARCHAR and LONG VARGRAPHIC data types	To be determined	<ul style="list-style-type: none"> • LONG VARCHAR and LONG VARGRAPHIC data types have been deprecated
maxagents and maxcagents configuration parameters	Version 9.5	<ul style="list-style-type: none"> • Some database manager configuration parameters have changed
Netscape browser support	Version 9.7	<ul style="list-style-type: none"> • Netscape browser support has been discontinued
numsegs configuration parameter	Version 9.5	<ul style="list-style-type: none"> • Some database configuration parameters have changed

Table 38. Functionality deprecated in Version 9.5 and possibly discontinued in a later release (continued)

Functionality	Discontinued in release	Links to additional information
query_heap_sz configuration parameter	Version 9.5	<ul style="list-style-type: none"> Some database manager configuration parameters have changed
sqluadcu API	Version 9.7	<ul style="list-style-type: none"> sqluadcu API is deprecated sqluadcu API and sql_authorization data structure have been discontinued
Static data stream snapshot output	To be determined	<ul style="list-style-type: none"> Static data stream snapshot output is deprecated
Subset of SQL administrative routines	To be determined	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> Web Object Runtime Framework (WORF) is deprecated Web Object Runtime Framework (WORF) support has been discontinued
XML Extender	Version 9.7	<ul style="list-style-type: none"> XML Extender is deprecated XML Extender has been discontinued
HP-UX 32-bit client support	To be determined	<ul style="list-style-type: none"> HP-UX 32-bit client support is deprecated
DB2 Health Advisor	To be determined	DB2 Health Advisor has been deprecated at http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.wn.doc/doc/i0059075.html

Table 39. 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	<ul style="list-style-type: none"> Some registry and environment variables are discontinued
DB2ATLD_PORTS	To be determined	<ul style="list-style-type: none"> Some registry and environment variables are discontinued
DB2_ASYNC_IO_MAXFILOP	To be determined	<ul style="list-style-type: none"> Some registry and environment variables are discontinued
DB2_BAR_AUTONOMIC_DISABLE	To be determined	<ul style="list-style-type: none"> Some registry and environment variables are discontinued
DB2BPVARS	To be determined	<ul style="list-style-type: none"> Some registry and environment variables are discontinued
DB2COUNTRY	To be determined	<ul style="list-style-type: none"> Some registry and environment variables are discontinued
DB2DEFPREP	To be determined	<ul style="list-style-type: none"> Some registry and environment variables are discontinued

Table 39. 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_DJ_COMM	To be determined	<ul style="list-style-type: none"> Some registry and environment variables are deprecated
DB2DMNBCKCLR	To be determined	<ul style="list-style-type: none"> Some registry and environment variables are discontinued
DB2FFDC	To be determined	<ul style="list-style-type: none"> Some registry and environment variables are discontinued
DB2_HASH_JOIN	To be determined	<ul style="list-style-type: none"> Some registry and environment variables are discontinued
DB2_INDEX_FREE	To be determined	<ul style="list-style-type: none"> Some registry and environment variables are discontinued
DB2_MAP_XML_AS_CLOB_FOR_DLC	To be determined	<ul style="list-style-type: none"> Some registry and environment variables are discontinued
DB2MEMMAXFREE	To be determined	<ul style="list-style-type: none"> Some registry and environment variables are discontinued
DB2_PARTITIONEDLOAD_DEFAULT	To be determined	<ul style="list-style-type: none"> Some registry and environment variables are discontinued
DB2PRIORITIES and DB2NTPRICLASS	To be determined	<ul style="list-style-type: none"> Some registry and environment variables are discontinued
DB2ROUTINE_DEBUG	To be determined	<ul style="list-style-type: none"> Some registry and environment variables are discontinued
DB2_RR_TO_RS	To be determined	<ul style="list-style-type: none"> Some registry and environment variables are discontinued
DB2_SNAPSHOT_NOAUTH	To be determined	<ul style="list-style-type: none"> Some registry and environment variables are discontinued
DB2_TRUSTED_BINDIN	To be determined	<ul style="list-style-type: none"> Some registry and environment variables are discontinued
DB2_UPDATE_PART_KEY	To be determined	<ul style="list-style-type: none"> Some registry and environment variables are discontinued
DB2_VENDOR_INI	To be determined	<ul style="list-style-type: none"> Some registry and environment variables are discontinued
DB2YIELD	To be determined	<ul style="list-style-type: none"> Some registry and environment variables are discontinued

Functionality deprecated in Version 9.7

Table 40. Functionality deprecated in Version 9.7

Functionality	Links to additional information
CREATE EVENT MONITOR FOR DEADLOCKS statement and DB2DETAILDEADLOCK event monitor	<ul style="list-style-type: none"> CREATE EVENT MONITOR FOR DEADLOCKS statement and DB2DETAILDEADLOCK event monitor have been deprecated
CREATE EVENT MONITOR FOR TRANSACTIONS event monitor	<ul style="list-style-type: none"> CREATE EVENT MONITOR FOR TRANSACTIONS event monitor has been deprecated

Table 40. Functionality deprecated in Version 9.7 (continued)

Functionality	Links to additional information
Reporting of metrics in details_xml by the statistics event monitor	<ul style="list-style-type: none"> Reporting of metrics in details_xml by the statistics event monitor has been deprecated
DB2 Governor and Query Patroller	<ul style="list-style-type: none"> DB2 Governor and Query Patroller have been deprecated
IBM DB2 Geodetic Data Management Feature	<ul style="list-style-type: none"> IBM DB2 Geodetic Data Management Feature has been deprecated
db2imigr , db2ckmig , and MIGRATE DATABASE commands; sqlmgdb and sqlgmdb APIs	<ul style="list-style-type: none"> Instance and database migration commands and APIs have been deprecated
db2iupdt command option -s	<ul style="list-style-type: none"> db2iupdt command -s option has been deprecated (Linux and UNIX)
-file option of the db2rfpen command	<ul style="list-style-type: none"> -file option of db2rfpen command has been deprecated
Worksheet Format (WSF) for Export and Load utilities	<ul style="list-style-type: none"> Worksheet Format (WSF) for Export and Load utilities has been deprecated
LIST TABLESPACES and LIST TABLESPACE CONTAINERS commands and related APIs: <ul style="list-style-type: none"> sqlbctsq sqlbftsq sqlbftpq sqlbgtss sqlbmstsq sqlbotsq sqlbstpq sqlbstsq sqlbtcq 	<ul style="list-style-type: none"> LIST TABLESPACES and LIST TABLESPACE CONTAINERS commands have been deprecated
MIGRATE_PRIOR_VERSIONS and the CONFIG_ONLY response file keywords	<ul style="list-style-type: none"> Some response file keywords have been deprecated
CONVERT parameter of the REORG INDEXES command	<ul style="list-style-type: none"> Type-1 indexes have been discontinued
DB2LOADQUERY_TYPE1_INDEXES and the DB2REORG_CONVERT values in certain DB2 API data structures	<ul style="list-style-type: none"> Type-1 indexes have been discontinued
sqlugtpi API	<ul style="list-style-type: none"> sqlugtpi API has been deprecated
sqlugrpn API	<ul style="list-style-type: none"> sqlugrpn API has been deprecated
db2HistoryCloseScan, db2HistoryGetEntry, db2HistoryOpenScan, and db2HistoryUpdate APIs	<ul style="list-style-type: none"> COBOL and FORTRAN language support has been deprecated for db2History APIs
DB2SE_USA_GEOCODER	<ul style="list-style-type: none"> DB2SE_USA_GEOCODER has been deprecated
Subset of Net Search Extender features and commands	<ul style="list-style-type: none"> Subset of Net Search Extender features and commands have been deprecated
Subset of SQL administrative routines	<ul style="list-style-type: none"> Deprecated Version 9.7 SQL administrative routines and their replacement routines or views
DB2 administration server (DAS)	<ul style="list-style-type: none"> DB2 administration server (DAS) has been deprecated

Table 40. Functionality deprecated in Version 9.7 (continued)

Functionality	Links to additional information
The following Control Center tools: <ul style="list-style-type: none"> • Activity Monitor • Command Editor • Configuration Assistant • Control Center and associated wizards and advisors • Control Center plug-in extensions • Event Analyzer • Health Center • Indoubt Transaction Monitor • Journal • License Center • Memory Visualizer • Query Patroller Center • Satellite Administration Center • Task Center 	<ul style="list-style-type: none"> • Control Center tools have been deprecated
DB2 Health Advisor	<ul style="list-style-type: none"> • DB2 Health Advisor has been deprecated
Health monitor	<ul style="list-style-type: none"> • Health monitor has been deprecated
HP-UX 32-bit client support	<ul style="list-style-type: none"> • HP-UX 32-bit client support has been deprecated
Visual Studio 2005	<ul style="list-style-type: none"> • Visual Studio 2005 support has been deprecated

Table 41. Registry variables deprecated in Version 9.7

Registry variables	Links to additional information
DB2_CAPTURE_LOCKTIMEOUT	<ul style="list-style-type: none"> • Some registry and environment variables have been deprecated
DB2_QP_BYPASS_APPLICATIONS	<ul style="list-style-type: none"> • Some registry and environment variables have been deprecated
DB2_QP_BYPASS_COST	<ul style="list-style-type: none"> • Some registry and environment variables have been deprecated
DB2_QP_BYPASS_USERS	<ul style="list-style-type: none"> • Some registry and environment variables have been deprecated
DB2_SERVER_ENCALG	<ul style="list-style-type: none"> • Some registry and environment variables have been deprecated
DB2_USE_DB2JCCT2_JROUTINE	<ul style="list-style-type: none"> • Some registry and environment variables have been deprecated

Part 4. Appendixes

Appendix A. DB2 Version 9.7 for Linux, UNIX, and Windows fix pack summary

Version 9.7 fix packs contain important technical changes and new functionality that might affect your product usage.

Review the technical changes and new functionality that are included in the following Version 9.7 fix packs. Each fix pack for a release is cumulative: it contains all the technical changes and new functionality that were included in previous fix packs for that release.

- “Fix Pack 6”
- “Fix Pack 5” on page 331
- “Fix Pack 4” on page 333
- “Fix Pack 3a” on page 336
- “Fix Pack 3” on page 336
- “Fix Pack 2” on page 337
- “Fix Pack 1” on page 340

Fix Pack 6

Fix Pack 6 contains the functionality of the previous fix packs and includes the following changes:

- The behavior of the DB2 call level interface (CLI) is changed when you specify both the **SchemaList** and **SchemaFilter** keywords with *USRLIBL value for connections to DB2 for i. For more information, see “SchemaFilter IBM Data Server Driver configuration keyword” and “SchemaList CLI/ODBC configuration keyword”.
- The default values of selected data server driver keywords that are associated with automatic client reroute is changed. For more information, see “Default values of selected automatic client reroute IBM Data Server Driver keywords have changed” on page 252.
- The default value or behavior of selected data server driver keywords have changed. For more information, see “Default value or behavior of selected IBM Data Server Driver keywords have changed” on page 252.
- The behavior of IBM .NET Data Provider is changed if you specify both the **SchemaFilter** keyword and the `DB2ConnectionStringBuilder.SchemaList` property with a *USRLIBL value for connections to DB2 for i. For more information, see “SchemaFilter IBM Data Server Driver configuration keyword” and “DB2ConnectionStringBuilder.SchemaList Property”.
- New logical groups have been added to the CREATE EVENT MONITOR statement for the statistics event monitor. For more information, see “FP6: New logical data groups added to the statistics event monitor” on page 55.

Fix Pack 6 also contains the following enhancements:

- Versions of the IBM Data Server Driver for JDBC and SQLJ that are included with this fix pack contain several enhancements. For more information, see “FP6: Driver versions enhancements” on page 147.
- Support was added for transactional control in compiled language SQL user-defined functions (UDFs). When you set the

DB2_COMPATIBILITY_VECTOR registry variable to support runtime routine SQL data-access-level enforcement, COMMIT and ROLLBACK statements are allowed in compiled PL/SQL UDFs and compiled SQL PL UDFs if the functions are defined with the MODIFIES SQL DATA clause using the CREATE FUNCTION statement.

- Restrictions on the Processor Value Unit (PVU) limit for DB2 Workgroup Server Edition were removed. For more information, see “Processor Value Unit pricing”.
- The **db2mscs** command now provides the **-user** and **-passwd** parameters to avoid specifying this information in the `db2mscs.cfg` file. For more information, see “db2mscs - Set up Windows failover utility command”.
- The SQL_BINARY and SQL_VARBINARY SQL data types are now supported with DB2 for i V6R1 and later. For more information, see “SQL data type representation in ADO.NET database applications”.
- You can now specify the XML data type when creating global variables, specifying parameters to create compiled SQL functions, or defining local XML variables in compiled SQL functions. For more information, see “FP6: XML data type support added in global variables and compiled SQL functions” on page 26.
- CLI now supports SQL_BINARY and SQL_VARBINARY data types for DB2 for i Version 6 Release 1 and later. For more information, see “SQL symbolic and default data types for CLI applications”.
- CLI now supports array input by using the SQL_ATTR_PARAMSET_SIZE statement attribute, inside a trusted procedure body. For more information, see “Statement attributes (CLI) list”.
- CLI now supports retrieval of the server time for COMMIT or ROLLBACK SQL operations on DB2 for z/OS Version 10 and later. For more information, see “Connection attributes (CLI) list”.
- CLI now supports the SQL_C_CURSORHANDLE C data type for use with the SQL_CURSORHANDLE SQL data type. For more information, see “SQL symbolic and default data types for CLI applications”.
- CLI now supports the **QueryTimeout** keyword in the `db2dsdriver.cfg` file. For more information, see “QueryTimeout IBM Data Server Driver configuration keyword”.
- You can specify LDAP variables using the data server driver keywords. For more information, see “Call level interface (CLI) functionality has been enhanced” on page 150.
- Windows system environment variables are now updated to include the IBM Data Server Driver Package path. For more information, see “FP6: Environment variables now contain the driver path (Windows)” on page 131.
- You can now check the status of NSE text indexes. For more information, see “FP6: New utility validates a Net Search Extender text index (Windows, AIX)” on page 131“FP6: Validate your Net Search Extender text index”.
- You can use the **SQLCODEMAP** Data Server Driver configuration keyword or the **SQLCODEMAP** CLI/ODBC keyword to specify whether SQLCODE is used or turned off. For more information, see “SQLCODEMAP IBM Data Server Driver Configuration Keyword”.
- IBM .NET Data Provider now supports changing a user password by using the `DB2ConnectionStringBuilder.NewPWD` property. For more information, see “DB2ConnectionStringBuilder.NewPWD Property”.
- IBM .NET Data Provider now supports setting the CURRENT SQLID special register on DB2 for z/OS servers by using the new **CurrentSQLID** IBM Data

Server Driver configuration keyword or the DB2ConnectionStringBuilder.CurrentSQLID property. For more information, see “DB2ConnectionStringBuilder.CurrentSQLID Property”.

- IBM .NET Data Provider now supports setting the **FetchBufferSize** IBM Data Server Driver configuration keyword to configure the buffer size that is used by fetch requests. For more information, see “FetchBufferSize IBM Data Server Driver configuration keyword”.
- You can use the **ZOSDBNameFilter** IBM Data Server Driver configuration keyword or DB2ConnectionStringBuilder.DBName property to filter the query result of DB2 for z/OS base tables. For more information, see “DB2ConnectionStringBuilder.DBName Property”.
- You can use the CacheUSRLIBLValue property to cache *USRLIBL for connections to DB2 for i V6R1 and later. For more information, see “DB2Connection.CacheUSRLIBLValue Property”.
- IBM .NET Data Provider now supports the **DelimIdent** keyword and provides a default value for the ClientWorkStation property. For more information, see “IBM Data Server Provider for .NET is enhanced” on page 160 “IBM Data Server Provider for .NET has been enhanced”.
- IBM .NET Data Provider now supports DB2DateTime structure for use with Informix database server. For more information, see DB2DateTime Structure DB2DateTime Structure.
- IBM entity provider supports DB2 and Informix functions for LINQ to Entities queries. For more information, see Provider support for Microsoft Entity Framework Provider support for Microsoft Entity Framework.
- New fields are added to the DB2Decimal class. For more information, see “IBM Data Server Provider for .NET is enhanced” on page 160 “IBM Data Server Provider for .NET has been enhanced”.
- You can use the db2dsdriver.cfg file to set the DIAGLEVEL keyword. For more information, see Diaglevel IBM Data Server Driver configuration keyword **Diaglevel** IBM Data Server Driver configuration keyword.
- New properties and methods are added to the DB2Blob and DB2Clob classes. For more information, see “IBM Data Server Provider for .NET is enhanced” on page 160 “IBM Data Server Provider for .NET has been enhanced”.
- You can use the ClearUSRLIBLCache method to clear the *USRLIBL cache for connections to DB2 for i V6R1 and later. For more information, see “DB2Connection.ClearUSRLIBLCache Method”.
- Enhancements to the **db2cli** command include new features to the **registerdsn** parameters in Windows. For more information, see “Call level interface (CLI) functionality has been enhanced” on page 150.
- The installDSDriver command on UNIX and Linux operating systems now creates the db2profile and db2cshrc files to set required environment variables. For more information, see “installDSDriver - Extract Data Server Driver components command”.
- Seamless automatic client reroute is enhanced. The CLI driver can now perform seamless failover when a COMMIT or ROLLBACK statement is issued against an unreachable server after all the data is returned. For more information, see “Call level interface (CLI) functionality has been enhanced” on page 150.
- Enhancements to CLI include support for client information properties against DB2 for i. For more information, see “Call level interface (CLI) functionality has been enhanced” on page 150.

- The threshold violation event monitor now captures more information about the application that violated a threshold. For more information, see “Monitoring threshold violations”.
- The statistics event monitor is enhanced with additional logical data groups that make it possible to view metrics monitor elements directly, instead of from XML documents. These new logical data groups present versions of each monitor element that shows the change in value of that monitor element since the last statistics collection or database activation, whichever was more recent. In addition, a new element was added to the output statistics event monitor that collects these metrics into a single XML document. For more information, see “FP6: New logical data groups added to the statistics event monitor” on page 55, “FP6: XML document metrics stores system metrics collected by statistics event monitor” on page 56.
- You can use three new table functions to retrieve system and network information with SQL queries. These functions are intended to replace the table function ENV_GET_SYS_RESOURCES and the administrative view ENV_SYS_RESOURCES, which are deprecated as of this fix pack. For more information, see “FP6: New table functions provide SQL-based access to system information” on page 54.
- The ADMIN_MOVE_TABLE stored procedure now has a REDIRECT option that forwards changes directly to the target table instead of capturing the changes in the staging table. For more information, see “ADMIN_MOVE_TABLE procedure - Move tables online”.
- You can now use an in-database approach to analytics in the data warehouse by running a SAS embedded process on the DB2 database server. For more information, see “In-database analytics with SAS embedded process support added” on page 109.
- You can specify the UOWTOTALTIME threshold value in multiples of 10 seconds. For more information, see “Time-based thresholds support finer granularity” on page 91.
- The **REBIND** command now has a **FUNCPATH** parameter that specifies the function path to resolve user-defined distinct types and functions in static SQL. For more information, see “REBIND command”.
- The **db2look** command has a new **-noimplschema** parameter. If you specify this parameter with the **-e** parameter, the CREATE SCHEMA DDL statements for implicitly created schemas are not generated. For more information, see .
- Embedded SQL applications can use timeout values specified in the db2dsdriver.cfg file. For more information, see “Use of the db2dsdriver.cfg configuration file by embedded SQL applications”.
- Embedded SQL applications can call a stored procedure using three-part name. For more information, see “Enabling compatibility features for migration”.
- Embedded SQL applications can call a stored procedure using the new syntax. For more information, see “Enabling compatibility features for migration”.
- Embedded SQL applications can use the **WHENEVER condition DO action** statement to take a specified action when an exception condition occurs. For more information, see “WHENEVER statement” .
- You can now use the **fcm_parallelism** configuration parameter to control the degree of parallelism that is used for communication between members within a DB2 instance. For more information, see “FCM parallelism support added” .

Fix Pack 5

Fix Pack 5 contains the functionality of the previous fix packs and includes the following changes:

- The IBM Software Development Kit for Java that is packaged with DB2 products now use the Java 6.0.9.1 version. This Java version is now the minimum level supported by DB2 products. It addresses a critical security vulnerability (CVE-2010-4476) that might cause the Java Runtime Environment to hang. For more information about Java versions that are bundled with DB2 products, see “Java software support for DB2 products”.
- On Windows operating systems, IBM Tivoli System Automation for Multiplatforms (SA MP) 3.2 is now bundled with the DB2 installation media. For Tivoli SA MP levels on all supported operating systems, see “Supported software and hardware for IBM Tivoli System Automation for Multiplatforms (SA MP)”.
- PL/SQL statement compilation and execution is now supported for DB2 Express-C. For a list of restrictions on PL/SQL support, see “Restrictions on PL/SQL support”
- The history file will no longer be exclusively locked during the automatic deletion of recovery objects or during prune operations which delete the physical files. For more information, see “FP5: History file no longer locked during automatic deletion of recovery objects” on page 235.
- If **DB2_RESTORE_GRANT_ADMIN_AUTHORITIES** is set to ON, restores to an existing database is automatically granted SECADM and DBADM authorities. For more information, see the "DB2_RESTORE_GRANT_ADMIN_AUTHORITIES" entry in “System environment variables” in *Database Administration Concepts and Configuration Reference*.
- The self-tuning memory manager (STMM) shared memory segment now has owner-read and owner-write permissions only. On systems with multiple instances, this introduces compatibility issues against databases belonging to downlevel instances where the **database_memory** configuration parameter is set to AUTOMATIC. If you do not upgrade all instances simultaneously and any of the upgraded instances are root installations, downlevel instances will be unable to have overall database memory usage tuned by STMM. To mitigate this problem, use the new **DB2STMM** registry variable to revert to the previous non-restrictive permissions for the STMM shared memory segment. For more information, see the "DB2STMM" entry in “Miscellaneous variables” in *Database Administration Concepts and Configuration Reference*.
- DB2 Geodetic Data Management Feature has been deprecated and might be removed in a future release. For more information, see “FP5: IBM DB2 Geodetic Data Management Feature software has been deprecated” on page 290.
- Microsoft Visual Studio 2005 support has been deprecated and might be removed in a future release. For more information, see “FP5: Microsoft Visual Studio 2005 support has been deprecated” on page 290.

Fix Pack 5 also contains the following enhancements:

- Versions of the IBM Data Server Driver for JDBC and SQLJ that are included with this fix pack contain several enhancements. For more information, see “FP5: Driver versions enhancements” on page 144.
- The new super asynchronous (SUPERASYNC) HADR synchronization mode which ensures that transactions can never be blocked or experience elongated

response times due to network interruptions or congestion. For more information, see “FP5: Super asynchronous HADR synchronization mode has been added” on page 64.

- You can now query inline values of XML and large object (LOB) types when you use the HADR reads on standby feature. For more information, see **HADR reads on standby feature** in the *Data Recovery and High Availability Guide and Reference*.
- The IBM COBOL for AIX Version 3.1 and Version 4.1 compilers are now supported for both 32-bit and 64-bit DB2 database application development. For more information, see “Support for database application development in COBOL”.
- Enhancements to facilitate the migration of embedded SQL C applications from other database systems. The following enhancements are included:
 - Support of local scope for host variables
 - The BREAK action in a WHENEVER statement
 - Use of a string literal to prepare a statement
 - The VARCHAR and int types
 - The sqlglm() function to retrieve the complete error message text

For more information, see “Host variable names in C and C++”, “Supported SQL data types in C and C++”, and “Enabling compatibility features for migration ” in *Developing Embedded SQL Applications*.

- The new **EXCLUDE**, **PRECHECK**, and **QUIESCE DATABASE** command parameters for the **REDISTRIBUTE DATABASE PARTITION GROUP** command make it easier to prevent errors and perform early error detection. For more information, see “FP5: Access control, error checking, and table exclusion during data redistribution have been improved” on page 19.
- IBM solidDB Universal Cache is now bundled with IBM Database Enterprise Developer Edition. See “FP5: IBM Database Enterprise Developer Edition product bundle extended” on page 190
- Additional diagnostic information that is available through the **db2pd** command makes it easier to troubleshoot load operations. You can obtain the diagnostic information by using the enhanced **-utilities** parameter and the new **-load** parameter. For more information, see “FP5: Load serviceability has been improved” on page 195.
- Examining databases for architectural correctness through the **db2dart** command is improved to increase its performance. The **/QCK** option and other options and actions are updated to help improve the performance of the **db2dart** command. For more information, see “FP5: db2dart command has extended functionality that helps improve performance” on page 196.
- First occurrence data collection (FODC) supports new manual collection types for problems related to processor usage, memory usage, and database connections. FODC now also automatically collects diagnostic data when a user-defined threshold condition is exceeded. For more information, see “FP5: First occurrence data collection supports new collection types and collection triggered by user-defined thresholds” on page 194.
- If you work with relational database products other than the DB2 product, Fix Pack 5 contains enhancements that make the DB2 product more familiar to you. For more information, see “FP5: SQL compatibility has been enhanced” on page 85.
- You can now use the **db2adutl** command to upload locally stored backup images to Tivoli Storage Manager (TSM). For more information, see “FP5: db2adutl enhancements simplify backup image and log file management” on page 63.

- Two new monitoring functions have been added to let you examine memory usage. For more information, see “FP5: New monitoring table functions return information about memory” on page 54.
- You can now invoke the DB2GSE.ST_REGISTER_SPATIAL_COLUMN stored procedure to calculate geographic extents for spatial data. You can display the geographic extent information by querying the DB2GSE.ST_GEOMETRY_COLUMNS catalog view. For more information, see “ST_REGISTER_SPATIAL_COLUMN stored procedure” in *Spatial Extender and Geodetic Data Management Feature User’s Guide and Reference*.
- You can specify the ACTIVITYTOTALTIME threshold value in multiples of 10 seconds. For more information, see “Time-based thresholds support finer granularity” on page 91.
- You can use table space modification information to make more informed decisions about the way that you perform backups. This information can be displayed through a new suboption for the **db2pd -tablespaces** command or a new MON_GET_TABLESPACE table function monitor element. For more information, see “FP5: Table space modification status can be checked” on page 193.
- Allocating trace resources for the trace facility (which you invoke with the **db2trc** command) for remote clients can help improve application performance. Also, you can trace the operations of clients that use a fenced procedure. For more information, see “FP5: db2trc command has been improved for clients” on page 194.
- The DB2 Design Advisor can now be called within an SQL statement using the new DESIGN_ADVISOR procedure. For more information, see the “DESIGN_ADVISOR procedure”.
- Troubleshooting problems that arise from unsuccessful upgrades is now made easier by support for collecting diagnostic data before the upgrade operation. You can collect the pre-upgrade data by specifying the new **-preupgrade** parameter for both the **db2fodc** command and the **db2support** command. You can collect additional data after the upgrade operation to help troubleshoot an instance creation problem by specifying the new **-c1p** parameter for the **db2fodc** command. For more information, see “FP5: Diagnosing upgrade problems is easier” on page 193.
- You can now use the COPY_USE_LOAD option to specify that the **ADMIN_MOVE_TABLE** procedure is to perform recoverable load operations. For more information, see “FP5: ADMIN_MOVE_TABLE stored procedure now supports recoverable load” on page 64.
- Merged backup images are now fully integrated with automatic recovery utilities and database history management infrastructure. For more information, see “FP5: Automatic recovery utilities now recognize merged backups” on page 63.
- Several administrative views and table functions for monitoring database systems have been updated. The affected routines are MON_BP_UTILIZATION, MON_GET_BUFFERPOOL, MON_GET_TABLE, and MON_GET_TABLESPACE. For more information, see “FP5: Some administrative routines and views have changed” on page 235.

Fix Pack 4

Fix Pack 4 contains the functionality of the previous fix packs and includes the following changes:

- Support to perform distributed installations using Microsoft Systems Management Server on Windows operating systems has been deprecated. For

more information, see “FP4: Distributed installation support with Microsoft Systems Management Server is deprecated (Windows)” on page 301.

- The DB2 Health Advisor has been deprecated. For more information, see “FP4: DB2 Health Advisor has been deprecated” on page 289.

Fix Pack 4 also contains the following enhancements:

- Versions of the IBM Data Server Driver for JDBC and SQLJ that are included with this fix pack contain several enhancements. For more information, see “FP4: Driver versions enhancements” on page 143.
- A new aggregate function, LISTAGG, has been added. The LISTAGG function aggregates a set of string elements into one string by concatenating the strings. Optionally, a separator string can be provided which is inserted between contiguous input strings. For more information, see **LISTAGG aggregate function** in the *SQL Reference, Volume 1*.
- You can use pattern expressions in a LIKE predicate which are based on other columns. The LIKE predicate can therefore be used as a fuzzy join predicate where the joined table provides the pattern. You can, for example, look up patterns such as area codes to filter telephone numbers from another table. For more information, see **LIKE predicate** in the *SQL Reference, Volume 1*.
- You can specify two-part names for tables and views with the **db2look** command to specify the schema of the table or view. In addition, the new **-xdep** and **-xddep** parameters generate authorization DDL statements (for example, GRANT statements) on the tables, specified by either the **-t** or **-tw** parameter, and their dependent objects. For more information, see “FP4: db2look command enhances DDL statement generation for database objects and dependents” on page 19.
- The WLM_COLLECT_STATS procedure now takes an input parameter called **wait**, which specifies that the procedure does not return until all the statistics have been written and flushed to the statistics event monitor tables. If you do not specify this parameter, the procedure returns immediately after initiating a statistics collection and reset. Monitoring tools can use the new functionality to collect WLM statistics in synchronous mode so that the tools are aware that all the data has been written to the statistics event monitor tables by the time the procedure returns. For more information, see “WLM_COLLECT_STATS procedure - Collect and reset workload management statistics” in *Administrative Routines and Views*.
- IBM Tivoli System Automation for Multiplatforms (SA MP) support enhanced for High Availability Disaster Recovery on Windows operating systems. New scripts have been added to improve the integration between DB2 High Availability Disaster Recovery (HADR) and IBM Tivoli System Automation for Multiplatforms (SA MP) on Windows operating systems. For more information, see “FP4: New scripts improve integration between DB2 High Availability Disaster Recovery and IBM Tivoli System Automation for Multiplatforms (Windows)” on page 62.
- Roving high availability (HA) failover support for partitioned database environments using the *N plus M* failover policy. For more information, see “FP4: Roving high availability (HA) failover reduces downtime in partitioned database environments” on page 63.
- You can use the **-recovery** option of the **db2pd** command to determine whether the catalog database partition has failed. For more information, see “db2pd - Monitor and troubleshoot DB2 database command” in the *Command Reference*.
- With the new **db2cklog** tool, you can now check the validity of archive log files before using these files during a rollforward recovery operation. For more information, see “FP4: Archive log files can be checked for validity” on page 201.

- You can now upgrade your DB2 Text Search server for quicker indexing and better search capabilities. For more information, see “FP4: DB2 Text Search infrastructure improvements” on page 201.
- The CREATE TRIGGER statement has changed. A trigger event clause can contain UPDATE, DELETE, and INSERT operations together in a single clause. Additionally, a BEFORE trigger can contain UPDATE, DELETE, INSERT, and modifying data routines in a compound SQL (compiled) statement. For more information, see “FP4: CREATE TRIGGER statement enhancements” on page 85.
- Support to perform distributed installations using Microsoft Systems Center Configuration Manager on Windows operating systems has been added. For more information, see “Installation of DB2 products using Microsoft Systems Center Configuration Manager (SCCM)” in *Installing DB2 Servers*.
- Diagnostic data logging has become more resilient. You can now set an alternate path for diagnostic data with the new **alt_diagpath** database configuration parameter. This parameter specifies an alternate path for diagnostic data logging that is used when the primary diagnostic path is unavailable. For more information, see “FP4: New configuration parameter reduces risk of losing diagnostic data” on page 200.
- The serviceability of large database systems has been improved. A number of functional enhancements have been made that address common pain points on large database systems, resulting in: Reduced amounts of accumulated diagnostic data; reduced overhead due to data collection on large systems; improvements to the accessibility of diagnostic data to service personnel; and, improvements to the ease of use of troubleshooting tools in complex systems. For more information, see “FP4: The serviceability of large database systems has improved” on page 196
- Access to activity metrics in the activity event monitor is simplified with the new activitymetrics logical data group. For more information, see “FP4: Simplified access to activity metrics in the activity event monitor” on page 53.
- Enhancements to facilitate the migration of embedded SQL C applications from other database systems. These compatibility features enable you to use C-array host variables and indicator variable arrays with FETCH INTO statements, extensions to CONNECT statement syntax, double quotation marks to specify file names with the INCLUDE statement, and the DYNAMIC_SQL option for the **BIND** command, to provide true dynamic SQL behavior. For more information, see “Enabling compatibility features for migration” in *Developing Embedded SQL Applications*.
- Enhancements to the DB2 Call Level Interface (CLI) include new command parameters, connection attribute, and functions. In addition, the db2diag.log path on Windows has changed and new log options are now available. For more information, see “Call level interface (CLI) functionality has been enhanced” on page 150.
- Enhancements to facilitate the migration of embedded SQL C applications from other database systems. These compatibility features enable you to use C-array host variables and indicator variable arrays with FETCH INTO statements, extensions to CONNECT statement syntax, double quotation marks to specify file names with the INCLUDE statement, and the **DYNAMIC_SQL** option for the **BIND** command, to provide true dynamic SQL behavior. For more information, see “Enabling compatibility features for migration” in *Developing Embedded SQL Applications*.
- The IBM XL C/C++ Enterprise Edition Version 11.0 compiler is now supported for DB2 database application development on the AIX platform. For more information, see “Support for database application development in C” and “Support for database application development in C++”.

- Support for .NET Framework has been enhanced. Starting in Version 9.7 Fix Pack 4, IBM Data Server Provider for .NET supports .NET Framework 4.0. For more information, see “Support for .NET development software”
- In Version 9.7 Fix Pack 4 and later fix packs, IBM Visual Studio Add-Ins support Visual Studio 2010. For more information, see “Supported IBM Data Servers and System Requirements”
- In Version 9.7 Fix Pack 4 and later fix packs, IBM Data Server Provider for .NET supports `FitHighPrecisionType` keyword. For more information, see the “FitHighPrecisionType Property”.
- In Version 9.7 Fix Pack 4 and later fix packs, IBM Data Server Provider for .NET and IBM Visual Studio Add-Ins no longer support U2 servers.
- New **installFixPack** command parameter enhances ability to apply fix packs. For more information, see “FP4: Installation fix pack support has been extended” on page 190.

Fix Pack 3a

Fix Pack 3a contains the functionality of the previous fix packs and includes the following change:

- For the DB2 Workgroup Server Edition the maximum allowed memory is increased from 16 GB to 64 GB.

Fix Pack 3a contains the following enhancement:

- Several advanced product capabilities are available in the DB2 database product DB2 Advanced Enterprise Server Edition for Linux, UNIX, and Windows. For more information, see “FP3: Advanced product capabilities” on page 3.

Fix Pack 3

Fix Pack 3 contains the functionality of the previous fix packs and includes the following changes:

- LOB strings of any length continue to be supported in comparisons using the LIKE predicate, NULL predicate, and the POSSTR function. LOB strings that have an actual length less than 32672 bytes are supported as operands in other predicates and the simple CASE expression.
- HP-UX 32-bit client support has been deprecated and might be discontinued in a future release. For more information, see “FP3: HP-UX 32-bit client support has been deprecated” on page 297.
- The settings of the **DB2_ITP_LEVEL** registry variable are ignored and have no effect on backup operations. For more information, see “Some registry and environment variables have changed” on page 225.
- Trap resilience functionality, a feature that keeps the instance active if certain traps occur, has now been extended to the load utility. For more information, see “Enhanced resilience to errors and traps reduces outages” on page 58.

Fix Pack 3 also contains the following enhancements:

- Versions of the IBM Data Server Driver for JDBC and SQLJ that are included with this fix pack contain several enhancements. For more information, see “FP3: Driver versions enhancements” on page 140.
- Support for the AIX 7.1 operating system. For more information, see “Installation requirements for DB2 servers and IBM data server clients (AIX)” in *Installing DB2 Servers*.

- You can specify that the system controller thread does not adjust resources below specific values by using the new `FCM_CFG_BASE_AS_FLOOR` option of the `DB2_FCM_SETTINGS` registry variable. For more information, see the `DB2_FCM_SETTINGS` entry in “Partitioned database environment variables” in the *Database Administration Concepts and Configuration Reference*.
- Improved support for target storage devices that support data deduplication. For more information, see “FP3: Data deduplication device support has been integrated into the backup utilities” on page 60.
- DB2 Text Search and Net Search Extender text indexes can now coexist on the same table column. For more information, see “FP3: DB2 Text Search and Net Search Extender index coexistence” on page 180.
- Range-clustered tables (RCT) are supported in a partitioned database environment. The distribution key must be a single column subset of the range-clustered table key. For more information, see “Restrictions on range-clustered tables”.
- A new tool `db2caem` (db2 Capture Activity Event Monitor data tool) has been created to simplify the process of capturing detailed diagnostic and runtime information about one or more statements. New **db2support** options have been added for the optimizer mode to collect the data captured by `db2caem`. For more information, see “FP3: Simplify capture of detailed statement information using the new `db2caem` tool with `db2support` integration options” on page 52.
- Two new features have been added to improve the granularity of the `db2trc` tool, these are an ability to trace only the members (or partitions) specified and an ability to trace based on a specific application ID (or application handle). For more information, see “FP3: Improvements to the granularity of the `db2trc` tool” on page 201.
- A new OLAP specification, `RATIO_TO_REPORT`, can be used to provide the ratio of a value compared to the sum of a group of values. For more information, see **OLAP specifications** in the *SQL Reference, Volume 1*.
- A new database configuration parameter, `CONNECT_PROC`, can be used to input a two part procedure name. This procedure is used as the connect procedure to customize the settings for the application environment when connecting to a database. For more information, see “FP3: Application environments can be customized during the connection process” on page 123.

Fix Pack 2

Fix Pack 2 contains the functionality of the previous fix pack and includes the following changes:

- If `DB2_RESTORE_GRANT_ADMIN_AUTHORITIES` is set to 0N, and you are restoring to a new database, then `SECADM`, `DBADM`, `DATAACCESS`, and `ACCESSCTRL` authorities are granted to the user that issues the restore operation. For more information, see the “`DB2_RESTORE_GRANT_ADMIN_AUTHORITIES`” entry in “System environment variables” in *Database Administration Concepts and Configuration Reference*.
- COBOL and FORTRAN language support for the `db2History` APIs has been deprecated and might be discontinued in a future release. For more information, see “FP2: COBOL and FORTRAN language support has been deprecated for `db2History` APIs” on page 297.
- The database manager uses a new formula to automatically adjust kernel parameter settings so that manual adjustments are not required to update Linux kernel parameters related to interprocess communication (IPC). For Version 9.7

Fix Pack 1 or earlier, you might still need to adjust your Linux kernel parameter settings. For more information, see “Kernel parameters requirements (Linux)” in *Installing DB2 Servers*.

- The DB2 Advanced Copy Services (ACS) component is no longer automatically installed during a compact installation. For more information, see “FP2: DB2 Advanced Copy Services (ACS) is not automatically included in a compact installation” on page 242.
- Issuing the **db2updv97** command is mandatory in order to execute the SYSPROC.ADMIN_GET_MSGS table function. However, after the **db2updv97** command is issued, if you are reverting back to Version 9.7 Fix Pack 1 or earlier, you are required to contact DB2 Support in order to have this table function run. For more information, see *db2updv97 - Update database to Version 9.7 fix pack*

Fix Pack 2 also contains the following enhancements:

- Versions of the IBM Data Server Driver for JDBC and SQLJ that are included with this fix pack contain several enhancements. For more information, see “FP2: Driver versions enhancements” on page 139.
- National character support has been extended by means of NCHAR, NCLOB, and NVARCHAR. For more information, see “National character strings” in *SQL Reference, Volume 1*.
- New scalar functions have been added to assist with national character support; NCHAR, NCLOB, NVARCHAR, TO_NCLOB, and TO_NCHAR. For more information, see “Supported functions and administrative SQL routines and views” in *SQL Reference, Volume 1*.
- Procedure enhancements introduced at Version 9.7 have been extended to user-defined functions (UDFs):
 - The ability to create UDFs and specify default values for parameters
 - The ability to invoke a UDF with named arguments

For more information, see “Default values and named arguments promote flexibility for creating and calling procedures and functions” on page 121.

- You can use the new WAIT FOR OUTCOME keyword in a SELECT statement to indicate the concurrent access resolution. WAIT FOR OUTCOME specifies to wait for the commit or rollback when encountering data in the process of being updated, deleted, or inserted. For more information, see “select-statement” in *Command Reference*.
- Support for applications on Solaris UltraSPARC and x64. For more information, see “Support for database application development in C” in *Getting Started with Database Application Development*.
- Tivoli Storage Manager (TSM) proxy nodes support. For more information, see “FP2: Proxy node support for the db2adutl command has been added” on page 60.
- Integrated support for POWER7 systems and the SUSE Linux Enterprise Server (SLES) 11 distribution. For more information, see “IBM Tivoli System Automation for Multiplatforms (SA MP) support has been improved” on page 188.
- The new **RESTRICTED ACCESS** option can be specified to prevent authorization checking for all connect attempts to the databases of a quiesced DB2 instance. The new option can also be used when there is a need to have exclusive connections to a database within the quiesced instance. For more information, see “FP2: New **RESTRICTED ACCESS** option restricts database connections within quiesced instance” on page 18.

- You can use the **RESTORE** command with the **TRANSPORT** option to copy table spaces and SQL schemas as a set from a database backup image to another active database. For more information, see “FP2: Databases can be restored using transportable sets” on page 61.
- You can obfuscate or encode the body of a routine, trigger, view, or PL/SQL package, so that the proprietary portion cannot be read by a user, but it is still understood by DB2 for Linux, UNIX, and Windows. For more information, see Obfuscation “Obfuscation” in *SQL Procedural Languages: Application Enablement and Support*.
- The **db2pd -reorgs index** command supports index reorg progress reporting for partitioned indexes. For more information, see “Additional system monitoring information can be generated” on page 49.
- The **db2pd** command has additional functionality to help control which engine dispatchable units (EDUs) are included in the output, the ability to define a time interval in some cases, and improved stack output on Linux operating systems. For more information, see “db2pd - Monitor and troubleshoot DB2 database” in *Command Reference*.
- Restrictions on the maximum size of core files (CORELIMIT) have been removed on AIX. For more information, see “db2pdcfg - Configure DB2 database for problem determination behavior ” in *Command Reference*.
- A new registry variable setting has been added to the **DB2_WORKLOAD** aggregate registry variable when it is set to SAP. For more information, see “Some registry and environment variables have changed” on page 225.
- The **DB2_SMS_TRUNC_TMPTABLE_THRESH** registry variable has a new default setting which prevents unnecessary file system access for small temporary objects while still truncating large temporary objects to 0 extents. For more information, see “Some registry and environment variables have changed” on page 225.
- The **WLM_SET_CONN_ENV** procedure enables for a particular connection the collection of activity data and measurement of section actuals (runtime statistics measured during section execution). For more information, see “WLM_SET_CONN_ENV” in *Administrative Routines and Views*.
- The **WLM_GET_CONN_ENV** table function returns for a particular connection the values of settings that control collection of activity data and section actuals. You can use this table function to check the current values of the settings applied by the **WLM_SET_CONN_ENV** stored procedure. For more information, see “WLM_GET_CONN_ENV” in *Administrative Routines and Views*.
- NULL indicators support has been extended to enable applications to use all-columns INSERT, UPDATE, and MERGE statements without having to specify the current value of columns whose values are not to be changed or inserted. For more information, see “References to host variables” in “Identifiers” in *SQL Reference, Volume 1*.
- Support for IBM Rational Developer for zSeries v7. For more information see “Support for database application development in COBOL” in *Getting Started with Database Application Development*.
- Transparent LDAP authentication and group lookup support on AIX is extended to certify Kerberos authentication support. For more information, see “FP1: Transparent LDAP authentication and group lookup is supported (Linux and UNIX)” on page 103.
- Auditing improvements have been added to allow for the replay of past database activities. For more information, see “FP2: Audit improvements allow replay of past database activities” on page 105.

- Two new table functions, `MON_GET_FCM` and `MON_GET_FCM_CONNECTION_LIST`, improve the monitoring of fast communications manager (FCM). For more information, see “FP2: Identify FCM issues more easily” on page 52.
- The IBM Data Server Provider for .NET has been enhanced in several ways that might help with application performance, data server compatibility, and simplifying application development. For more information, see “IBM Data Server Provider for .NET is enhanced” on page 160.
- Some queries using spatial data run faster in partitioned database environments. For more information, see “FP2: Materialized query tables with spatial columns can be replicated” on page 78.
- When using the `ADMIN_MOVE_TABLE` procedure, you can use the new `LOAD_MSGPATH` option to define the load message file path. The `FORCE` option no longer needs to be specified with the `COPY_USE_LOAD` option. For more information see “ADMIN_MOVE_TABLE procedure - Move an online table” in *Administrative Routines and Views*.
- Restrictions on the `DB2_SKIPDELETED` registry variable during an online `ADMIN_MOVE_TABLE` procedure have been removed. For more information see “ADMIN_MOVE_TABLE procedure - Move an online table” in *Administrative Routines and Views*.

Fix Pack 1

Fix Pack 1 includes the following changes:

- The `-file` option of the `db2rftp` command has been deprecated. For more information, see “FP1: -file option of db2rftp command has been deprecated” on page 296.
- The process to detach a data partition from a data partitioned table has been changed. For more information, see “FP1: Detach operation for data partitions has been changed” on page 233.
- If an XML schema that is registered in the DB2 XSR uses the `maxOccurs` attribute where the value is greater than 5000, the `maxOccurs` attribute value is treated as if you specified “unbounded”. For more information, see “FP1: XML schema `maxOccurs` attribute values greater than 5000 are parsed differently” on page 234.
- The workload management statistics collection interval is synchronized relative to a day of the week and an hour of the day, rather than relative to when the DB2 instance was started. For more information, see “FP1: Collection interval time for workload management statistics has changed” on page 224.

Fix Pack 1 also contains the following enhancements:

- Versions of the IBM Data Server Driver for JDBC and SQLJ that are included with this fix pack contain several enhancements. For more information, see “FP1: Driver versions enhancements” on page 138.
- Read operations support on High Availability and Disaster Recovery (HADR) standby databases. For more information, see “FP1: Read operations on HADR standby databases are supported” on page 59.
- DB2 Advanced Copy Services (ACS) support for the AIX 6.1 operating system. For more information, see “FP1: DB2 Advanced Copy Services (ACS) is supported on AIX 6.1” on page 58.
- Last referenced date support for some objects, which helps you understand when these were last used. For more information, see “FP1: Last referenced date is available for tables, table partitions, indexes, and packages” on page 45.

- The SUBSTRB scalar function, which returns a substring of a string. For more information, see “SUBSTRB scalar function” in *SQL Reference, Volume 1*.
- Compiled user defined functions with OUT and INOUT parameters are supported in SQL PL. For more information, see “SQL PL functionality has been extended for user-defined functions” on page 171.
- Global variables assignments in nested contexts are supported. For more information, see “FP1: Global variable assignments in nested contexts are supported” on page 173.
- OUT and INOUT parameters are supported in user-defined functions. For more information, see “FP1: User-defined functions support OUT and INOUT parameters” on page 130.
- Support for PL/SQL functions that modify the database. For more information, see “CREATE FUNCTION statement (PL/SQL)” in *SQL Procedural Languages: Application Enablement and Support*.
- The IBM Data Server Provider for .NET includes multiple enhancements. For more information, see “IBM Data Server Provider for .NET is enhanced” on page 160.
- The **db2pd** command has a new parameter that makes it easier to collect the fenced routines history information. For more information, see “FP1: Fenced routines history information is easier to collect” on page 203.
- The DB2 PL/SQL compiler supports FORALL and BULK COLLECT INTO syntax. For more information, see “FORALL statement (PL/SQL)” and “BULK COLLECT INTO clause (PL/SQL)” in *SQL Procedural Languages: Application Enablement and Support*.
- You can use new XQuery functions to retrieve current date and time values using the local time zone of the DB2 database system. For more information, see “FP1: XQuery functions make it easier to retrieve date and time values for local time zones” on page 31.
- The **diagpath** database manager configuration parameter has new values, which allow you to store DB2 diagnostic data in separate directories named according to the physical host, database partition, or both. The **db2diag** command also has a new **-merge** parameter to merge multiple **db2diag** log files. For more information, see “FP1: Diagnostic data can be stored in separate directories” on page 202.
- The new package cache event monitor captures information about cached statement entries after they have been flushed from the database package cache, which can help to resolve SQL query performance and problem determination issues. For more information, see “FP1: New event monitor for dynamic and static SQL statements in package cache” on page 48.
- New lock-related relational monitoring interfaces replace deprecated snapshot interfaces. For more information, see “FP1: New relational monitoring interfaces for locking events” on page 40.
- Runtime statistics are available for access plan operators. For more information, see “FP1: Explain enhanced with actual values for operator cardinality” on page 48.
- Section explain functionality captures explain information about a statement using only the contents of the runtime section. For more information, see “FP1: Statements from a runtime section can be explained” on page 47.
- New component time monitor elements can be combined with existing wait times monitor elements, provided in DB2 Version 9.7, to provide a

comprehensive breakdown of time spent within the DB2 database manager. For more information, see “Time-spent monitor elements are more comprehensive” on page 44.

- Time monitor elements, reported in XML documents, can be displayed and analyzed in a generic fashion using new row-based formatting functions. For more information, see “FP1: Table functions for row-based formatting of monitoring information are available” on page 51.
- Package cache information can be retrieved in XML form using a new package cache details table function. For more information, see “MON_GET_PKG_CACHE_STMT_DETAILS” in *Administrative Routines and Views*.
- New administrative views encapsulate key queries using the new monitoring table functions introduced in DB2 Version 9.7 and V9.7 Fix Pack 1. For more information, see “FP1: Monitoring table functions information can be viewed using administrative views” on page 50.
- A list of packages used within each unit of work can be obtained through the unit of work event monitor. For more information, see “A new unit of work event monitor supports transaction monitoring” on page 43.
- Reorganization of data or indexes for a specific data partition of a data partitioned table. For more information, see “FP1: Data partitions and partitioned indexes can be reorganized” on page 17.
- A partitioned table remains available during roll-out operations. For a partitioned table, a roll-out operation no longer takes the table offline. For more information, see “FP1: Partitioned table data remains available during roll-out operations” on page 77.
- Multidimensional clustering (MDC) block indexes are partitioned when creating a table that uses both MDC and table partitioning. For more information, see “Partitioned indexes on partitioned tables improve performance” on page 27.
- Distribution statistics are collected for indexes over XML data. For more information, see “FP1: Distribution statistics collected for XML columns” on page 34.
- The ADMIN_MOVE_TABLE procedure has new options which prevent locking overhead on the target table in the copy and swap phases and improve the data movement speed. For more information, see “Table data can be moved online using a new stored procedure” on page 14.
- Additional keywords can be added to the **db2relocatedb** command configuration file which makes it easier to relocate a database when the paths used are different. For more information, see “FP1: relocating databases using the db2relocatedb command has been improved” on page 18.
- New routines, views, and modules for monitoring, workload management, and explaining statements have been added and some routines have been changed. For more information, see “Some system catalog views, system-defined administrative routines and views have been added and changed” on page 254.
- You can monitor the progress of the **RUNSTATS** command as well as table and index reorganizations. For more information, see “Additional system monitoring information can be generated” on page 49.
- Transparent LDAP is supported on the Linux, HP-UX, and Solaris operating systems. For more information, see “FP1: Transparent LDAP authentication and group lookup is supported (Linux and UNIX)” on page 103.
- 32-bit GSKit libraries are now installed automatically. For more information, see “FP1: 32-bit GSKit libraries are included in the 64-bit DB2 product installation” on page 105.

- Additional support is provided for the GB18030 code set. For more information, see “GB18030 code set support has been extended” on page 191.
- DB2 database products installed on HP-UX operating systems now support long host names. For more information, see “Installation requirements for DB2 servers and IBM data server clients (HP-UX)” in *Installing DB2 Servers*.
- Multiple result sets can now be returned from an SQL procedure by enabling multiple instances of the same cursor. For more information, see “Returning result sets from SQL procedures” in *SQL Procedural Languages: Application Enablement and Support*.
- The **db2support** tool includes new filtering options that you can use to gather specific diagnostic data more easily and an archiving option for storing diagnostic files in a different location. For more information, see “FP1: db2support tool has been enhanced” on page 203.
- Work action sets can be defined at the workload level to control workloads, based on the type and size of work, before they enter the system. For more information, see “FP1: Work action sets can be defined at workload level” on page 94.
- The UOWTOTALTIME threshold specifies the maximum amount of time that a unit of work may spend in the DB2 engine. For more information, see “FP1: New time threshold limits unit of work duration” on page 94.
- A sample script (qpwlmmig.pl) has been provided to facilitate the migration from the deprecated DB2 Query Patroller environment to the DB2 workload manager environment. For more information, see “FP1: Script facilitates migration from Query Patroller to workload manager” on page 95.
- A new optional **AUTOGRANT** command parameter for the DB2 Text Search **ENABLE DATABASE FOR TEXT** command, which tries to grant the necessary DBADM with DATAACCESS privileges to the instance owner when running the **ENABLE** command, in case the instance owner misses these privileges for this database. For more information, see “db2ts ENABLE DATABASE FOR TEXT command” in the *Command Reference*.
- For Linux operating systems, users can specify that activated databases are to use minimal processing resources when the database manager is idle, using a new registry variable **DB2_MIN_IDLE_RESOURCES**. For more information, see the “DB2_MIN_IDLE_RESOURCES” entry in “Miscellaneous variables” in the *Database Administration Concepts and Configuration Reference*.
- With the new **DB2_USE_FAST_PREALLOCATION** registry variable, the Veritas fast allocation file system feature can be used to reserve table space, and speed up the process of creating or altering large table spaces and database restore operations. For more information, see the “DB2_USE_FAST_PREALLOCATION” entry in “Miscellaneous variables” in the *Database Administration Concepts and Configuration Reference*.
- With the new **DB2TCP_CLIENT_KEEPALIVE_TIMEOUT** registry variable, users can specify a keep alive setting that is lower than the system default, allowing the database manager to detect connection failures sooner. For more information, see the “DB2TCP_CLIENT_KEEPALIVE_TIMEOUT” entry in “Communications variables” in the *Database Administration Concepts and Configuration Reference*.
- The **DB2_WORKLOAD** aggregate registry variable now has a new value, **INFOR_ERP_LN**, which configures a set of registry variables for Infor ERP Baan. For more information, see the “DB2_WORKLOAD” entry in “System environment variables” in the *Database Administration Concepts and Configuration Reference*.

Appendix B. 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 email 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/e-business/linkweb/publications/servlet/pbi.wss. English Version 9.7 manuals in PDF format can be downloaded from www.ibm.com/support/docview.wss?uid=swg27015148 and translated DB2 manuals in PDF format can be downloaded from www.ibm.com/support/docview.wss?uid=swg27015149.

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 42. DB2 technical information

Name	Form Number	Available in print	Last updated
<i>Administrative API Reference</i>	SC27-2435-03	Yes	July, 2012
<i>Administrative Routines and Views</i>	SC27-2436-03	No	July, 2012
<i>Call Level Interface Guide and Reference, Volume 1</i>	SC27-2437-03	Yes	July, 2012
<i>Call Level Interface Guide and Reference, Volume 2</i>	SC27-2438-03	Yes	July, 2012
<i>Command Reference</i>	SC27-2439-03	Yes	July, 2012
<i>Data Movement Utilities Guide and Reference</i>	SC27-2440-01	Yes	July, 2012
<i>Data Recovery and High Availability Guide and Reference</i>	SC27-2441-03	Yes	July, 2012
<i>Database Administration Concepts and Configuration Reference</i>	SC27-2442-03	Yes	July, 2012
<i>Database Monitoring Guide and Reference</i>	SC27-2458-03	Yes	July, 2012
<i>Database Security Guide</i>	SC27-2443-02	Yes	July, 2012
<i>DB2 Text Search Guide</i>	SC27-2459-03	Yes	July, 2012
<i>Developing ADO.NET and OLE DB Applications</i>	SC27-2444-02	Yes	July, 2012
<i>Developing Embedded SQL Applications</i>	SC27-2445-02	Yes	July, 2012
<i>Developing Java Applications</i>	SC27-2446-03	Yes	July, 2012
<i>Developing Perl, PHP, Python, and Ruby on Rails Applications</i>	SC27-2447-02	No	July, 2012
<i>Developing User-defined Routines (SQL and External)</i>	SC27-2448-02	Yes	July, 2012
<i>Getting Started with Database Application Development</i>	GI11-9410-02	Yes	July, 2012

Table 42. DB2 technical information (continued)

Name	Form Number	Available in print	Last updated
<i>Getting Started with DB2 Installation and Administration on Linux and Windows</i>	GI11-9411-00	Yes	August, 2009
<i>Globalization Guide</i>	SC27-2449-00	Yes	August, 2009
<i>Installing DB2 Servers</i>	GC27-2455-03	Yes	July, 2012
<i>Installing IBM Data Server Clients</i>	GC27-2454-02	No	July, 2012
<i>Message Reference Volume 1</i>	SC27-2450-01	No	August, 2009
<i>Message Reference Volume 2</i>	SC27-2451-01	No	August, 2009
<i>Net Search Extender Administration and User's Guide</i>	SC27-2469-02	No	September, 2010
<i>Partitioning and Clustering Guide</i>	SC27-2453-02	Yes	July, 2012
<i>pureXML Guide</i>	SC27-2465-02	Yes	July, 2012
<i>Query Patroller Administration and User's Guide</i>	SC27-2467-00	No	August, 2009
<i>Spatial Extender and Geodetic Data Management Feature User's Guide and Reference</i>	SC27-2468-02	No	July, 2012
<i>SQL Procedural Languages: Application Enablement and Support</i>	SC27-2470-03	Yes	July, 2012
<i>SQL Reference, Volume 1</i>	SC27-2456-03	Yes	July, 2012
<i>SQL Reference, Volume 2</i>	SC27-2457-03	Yes	July, 2012
<i>Troubleshooting and Tuning Database Performance</i>	SC27-2461-03	Yes	July, 2012
<i>Upgrading to DB2 Version 9.7</i>	SC27-2452-03	Yes	July, 2012
<i>Visual Explain Tutorial</i>	SC27-2462-00	No	August, 2009
<i>What's New for DB2 Version 9.7</i>	SC27-2463-03	Yes	July, 2012
<i>Workload Manager Guide and Reference</i>	SC27-2464-03	Yes	July, 2012
<i>XQuery Reference</i>	SC27-2466-01	No	November, 2009

Table 43. DB2 Connect-specific technical information

Name	Form Number	Available in print	Last updated
<i>Installing and Configuring DB2 Connect Personal Edition</i>	SC27-2432-03	Yes	July, 2012
<i>Installing and Configuring DB2 Connect Servers</i>	SC27-2433-03	Yes	July, 2012
<i>DB2 Connect User's Guide</i>	SC27-2434-02	Yes	September, 2010

Table 44. Information Integration technical information

Name	Form Number	Available in print	Last updated
<i>Information Integration: Administration Guide for Federated Systems</i>	SC19-1020-02	Yes	August, 2009
<i>Information Integration: ASNCLP Program Reference for Replication and Event Publishing</i>	SC19-1018-04	Yes	August, 2009
<i>Information Integration: Configuration Guide for Federated Data Sources</i>	SC19-1034-02	No	August, 2009
<i>Information Integration: SQL Replication Guide and Reference</i>	SC19-1030-02	Yes	August, 2009
<i>Information Integration: Introduction to Replication and Event Publishing</i>	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:

Procedure

- 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 websites:
 - The IBM directory of world wide contacts at www.ibm.com/planetwide
 - The IBM Publications website 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 345.

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.

Procedure

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 10.1 topics, the *DB2 Information Center* URL is <http://publib.boulder.ibm.com/infocenter/db2luw/v10r1>

For DB2 Version 9.8 topics, the *DB2 Information Center* URL is <http://publib.boulder.ibm.com/infocenter/db2luw/v9r8/>.

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 9.1 topics, the *DB2 Information Center* URL is <http://publib.boulder.ibm.com/infocenter/db2luw/v9/>.

For DB2 Version 8 topics, go to the *DB2 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.

Procedure

- 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. 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. 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

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

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.

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.

Procedure

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 `update-ic` script:
`update-ic`
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 `update-ic.bat` file:

update-ic.bat

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**.

Procedure

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 with using DB2 database products.

DB2 documentation

Troubleshooting information can be found in the *Troubleshooting and Tuning Database Performance* or the Database fundamentals section of the *DB2 Information Center*. The troubleshooting information contains topics that can help you isolate and identify problems with DB2 diagnostic tools and utilities. There are also solutions to some of the most common problems and advice on how to solve problems you might encounter with your DB2 database products.

IBM Support Portal

See the IBM Support Portal 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 IBM Support Portal at http://www.ibm.com/support/entry/portal/Overview/Software/Information_Management/DB2_for_Linux,_UNIX_and_Windows.

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 C. 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 about 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.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 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, changes, or both 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those

websites. The materials at those websites are not part of the materials for this IBM product and use of those websites 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.

Licenses 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
U59/3600
3600 Steeles Avenue East
Markham, Ontario L3R 9Z7
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 or registered trademarks of Oracle, its affiliates, 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

Special characters

- global option deprecated 302
- .NET
 - enhancements 161
 - merge modules
 - packaging simplified 271
 - trusted context support added 149

Numerics

- 64-bit server 105

A

- access plans
 - reusing
 - overview 68
- ACCESSCTRL (access control) authority
 - overview 98
- act_remapped_in monitor element 92
- act_remapped_out monitor element 92
- activity event monitor
 - activity metrics 53
- Activity Monitor
 - deprecated 277
- ADMIN_EST_INLINE_LENGTH function
 - overview 31, 75
- ADMIN_IS_INLINED function
 - overview 31, 75
- ADMIN_MOVE_TABLE
 - procedure
 - changes 64
- ADMIN_MOVE_TABLE procedure
 - overview 14
- administration
 - modifications summary 221
- administration notification log
 - size control 60, 204
- administrative routines
 - additions 13, 254
 - changes 254
- administrative views
 - additions 13, 254
 - changes 254
 - monitoring information 50
- AES algorithm
 - alternate_auth_enc configuration parameter overview 100
- AGGSQLETEMPSPACE threshold
 - overview 91
- aliases
 - enhancements 121
 - public 121
- alt_diagpath configuration parameter
 - overview 200
- ALTER TABLE statement
 - ALTER COLUMN SET DATA TYPE enhancement 114
 - RENAME COLUMN clause 110
- alternate_auth_enc configuration parameter
 - overview 100, 222
- analytics
 - in-database 109
- anchored data types
 - overview 173
- annotated XML schema decomposition
 - enhancements 32
- APIs
 - deprecated 292
- applheapsz configuration parameter
 - change 238
- application development
 - enhancements summary 107
 - modifications summary 252
 - new samples summary 127
- application environment 123
- application handle
 - db2trc command 201
 - trace command 201
- application id
 - db2trc command 201
 - trace command 201
- applications
 - new samples summary 127
- array data types
 - associative array
 - overview 175
- assignments
 - global variable
 - nested contexts 173
- associative array data types
 - overview 175
- audit
 - replaying past database activity 105
- AUDIT_ARCHIVE stored procedure and table function
 - EXECUTE privilege enhancement 248
- AUDIT_DELM_EXTRACT stored procedure
 - EXECUTE privilege enhancement 248
- AUDIT_LIST_LOGS table function
 - EXECUTE privilege enhancement 248
- authentication
 - enhancement 100
- authentication configuration parameter
 - changes 222
- authorizations
 - model changes 98
- auto_reval database configuration parameter
 - overview 238
- automatic management scripts
 - SA MP and HADR (Windows) 62
- automatic revalidation
 - overview 112
- automatic storage databases
 - dropping storage paths
 - overview 10
 - enhancement 10
- automatic storage table spaces
 - rebalancing enhancement 10
 - reclaimable storage enhancement 11
- autonomous transactions
 - overview 123

B

- backups
 - enhancements summary 57
- blocknonlogged database configuration parameter
 - overview 238
- books
 - ordering 348
- Boolean data type
 - overview 174
- built-in functions
 - additions 254
 - changes 254
- built-in routines
 - additions 254
 - changes 254
- BULK COLLECT INTO clause
 - PL/SQL 273

C

- call level interface (CLI)
 - applications
 - enhancements 150
 - dynamic packages binding enhancement 150
 - enhancements 150
 - merge modules
 - packaging simplified 271
- casting
 - implicit 119
- catalog statistics
 - distribution statistics on XML columns 34
- CATALOG TCP/IP MODE command
 - enhancement 101
- catalog views
 - additions 254
 - changes 254
- changed functionality
 - summary 219, 221
- CHAR scalar function
 - return behavior altered 265
- checknseindex command
 - enhancement 131
- clusters
 - managing
 - Solaris SPARC support 58
- columns
 - renaming 110
- Command Editor
 - deprecated 277
- Command Line Processor Plus (CLPPlus)
 - overview 81
- commands
 - db2cklog
 - overview 201
 - db2ckupgrade
 - overview 299
 - db2iupgrade
 - overview 299
 - db2look
 - DDL statement generation enhancement 19
 - deprecated
 - Control Center 277
 - DB2 Governor 286
 - LIST TABLESPACE CONTAINERS 292
 - LIST TABLESPACES 292
 - Query Patroller 286

- commands (*continued*)
 - DESCRIBE
 - enhancement 17, 29
 - output changes 233
 - UPGRADE DATABASE
 - overview 299
- common SQL API stored procedures
 - overview 125
- compiled compound statements
 - overview 170
- compiled SQL functions
 - XML function parameters 26
- components
 - name changes 3
- compression
 - enhancements summary 5
 - index
 - overview 7
 - XML documents
 - overview 5, 35
- concurrency
 - scan sharing overview 71
- CONCURRENTDBCOORDACTIVITIES aggregate threshold change 232
- CONFIG_ONLY response file keyword 300
- Configuration Assistant (CA)
 - deprecated 277
- configuration parameters
 - enhancements 101
- connection string parameters
 - SSL support 101
- Control Center
 - Control Center
 - Optim tools comparison 277
 - deprecated tools 277
 - extensions
 - deprecated 277
 - mapping to Optim tools 277
- CPUTIME activity threshold
 - overview 91
- CREATE EVENT MONITOR FOR DEADLOCKS statement
 - deprecated 298
- CREATE EVENT MONITOR FOR TRANSACTIONS statement
 - deprecated 298
- CREATE INDEX statement
 - new default 221
- CREATE statement
 - OR REPLACE clause addition 110
- CREATE TRIGGER statement
 - BEFORE trigger 85
 - event predicates 85
- CREATE with errors
 - overview 111
- created temporary tables
 - overview 116
- cur_commit database configuration parameter
 - overview 238
- currently committed semantics
 - enhancement 70
- cursor data types
 - overview 175
- cursor stability (CS)
 - default behavior changed 253
 - enhancement 70
- cursor variables
 - overview 175

cursors
 parameter support added 175
customizing settings during the connection process 123

D

DAS
 deprecated 286

data
 distribution
 map size increased 15
 redistribution
 enhancements 19
 storage enhancements summary 5

data deduplication devices
 backup utilities 60

data dictionary
 Oracle
 compatible views 82

data types
 anchored
 overview 173

 associative array
 overview 175

 Boolean 174

 CLI enhancements 150

 cursor
 overview 175

 DATE 83

 LONG VARCHAR
 deprecated 291

 LONG VARGRAPHIC
 deprecated 291

 NUMBER 83

 row 176

 setting
 ALTER TABLE statement 114

 SQL PL 173

 VARCHAR2 83

data warehouse applications
 scalability improvement 15

DATAACCESS (data access) authority
 overview 98

database manager configuration parameters
 changed 222
 new 222

database objects
 modules 169

database partition servers
 enhancement 16

database-managed space (DMS)
 reclaimable storage overview 11

databases
 outage resiliency enhanced 58
 setup changes summary 236
 transporting schemas
 overview 61

DATE data type
 data type arithmetic support 83

date_compat database configuration parameter
 overview 238

DB2 administration server
 deprecated 286

DB2 Advanced Copy Services (ACS)
 installing
 compact type 242
 response file 242

DB2 Advanced Copy Services (ACS) (*continued*)
 supported operating systems 58

DB2 Advanced Enterprise Server Edition 3

DB2 Connect
 changes summary 207
 database manager configuration parameters
 changes 222
 enhancements
 summary 207

DB2 Connect Unlimited Edition (System z)
 license activation process 189

DB2 embedded application server (EAS)
 discontinued 309

DB2 Everyplace
 discontinued functionality 309

DB2 Express Edition
 licensing changes 237

DB2 Geodetic Data Management Feature
 discontinued 290

DB2 Governor
 deprecated 286

DB2 Health Advisor
 deprecated functionality 289

DB2 High Availability Disaster Recovery
 setting up automatic management scripts (Windows) 62

DB2 Information Center
 languages 350
 updating 351, 352
 versions 349

DB2 products
 availability 3
 packaging 3

DB2 Text Search 180
 authorizations
 changes 250
 commands
 authorization changes 250
 infrastructure improvements 201
 installing
 changes 242
 procedure authorization changes 250
 stored procedure authorization changes 250

DB2 Workgroup Edition
 licensing changes 237

DB2 workload management
 enhancements
 time-based thresholds granularity 91

DB2 workload manager
 changes
 statistics collection interval 224

CONCURRENTDBCOORDACTIVITIES threshold
 change 232

CONCURRENTDBCOORDACTIVITIES threshold
 enhancement 232

enhancements
 application-specific threshold controls 88
 buffer pool I/O priority 90
 high watermarks 89
 integration with Linux WLM 90
 IP address support 88
 Linux WLM support 90
 migration script 95
 monitoring 89
 priority aging 92
 resource controls 90
 service class tiering 92
 statistics collection 89

- DB2 workload manager *(continued)*
 - enhancements *(continued)*
 - summary 87
 - threshold actions 92
 - thresholds 88, 91, 95
 - unit of work time threshold control 95
 - wild card support 88
 - work action sets 94
 - workloads 88
 - licensing changes 237
 - migration script
 - Query Patroller to workload manager 95
 - statistics collection interval
 - synchronization 224
 - thresholds
 - AGGSQLTEMPSPACE 91
 - CPUTIME 91
 - SQLROWSREAD 91
 - work action sets
 - workload level 94
- DB2 XQuery functions
 - current-local-date
 - overview 31
 - current-local-dateTime
 - overview 31
 - current-local-time
 - overview 31
 - local-timezone
 - overview 31
- DB2_ATS_ENABLE registry variable
 - overview 225
- DB2_BACKUP_USE_DIO registry variable
 - overview 225
- DB2_CAPTURE_LOCKTIMEOUT registry variable
 - deprecated functionality 295
- DB2_COMPATIBILITY_VECTOR registry variable
 - enhancement 84
- DB2_DDL_SOFT_INVALID registry variable
 - overview 225
- DB2_DEFERRED_PREPARE_SEMANTICS registry variable
 - overview 225
- DB2_EVALUNCOMMITTED registry variable
 - changes 225
- DB2_EVMON_STMT_FILTER registry variable
 - new values 225
- DB2_FCM_SETTINGS registry variable
 - new values 225
- DB2_FORCE_OFFLINE_ADD_PARTITION environment variable
 - overview 225
- DB2_HADR_ROS registry variable
 - overview 225
- DB2_ITP_LEVEL registry variable
 - changes 225
- DB2_LIMIT_FENCED_GROUP registry variable
 - overview 225
- DB2_LOGGER_NON_BUFFERED_IO registry variable
 - changed default values 225
- DB2_NCHAR_SUPPORT registry variable
 - overview 225
- DB2_PMAP_COMPATIBILITY registry variable
 - overview 225
- DB2_PMODEL_SETTINGS registry variable
 - overview 225
- DB2_RESTORE_GRANT_ADMIN_AUTHORITIES registry variable
 - changes 225
- DB2_SERVER_ENCALG registry variable
 - changes 225
 - deprecated functionality 295
- DB2_SKIPDELETED registry variable
 - changes 225
- DB2_SKIPINSERTED registry variable
 - changes 225
- DB2_SQLROUTINE_PREPOPTS registry variable
 - new values 225
- DB2_SQLWORKSPACE_CACHE registry variable
 - overview 225
- DB2_STANDBY_ISO registry variable
 - overview 225
- DB2_THREAD_SUSPENSION variable
 - discontinued 312
- DB2_USE_DB2CCT2_JROUTINE registry variable
 - deprecated functionality 295
- DB2_USE_FAST_PREALLOCATION registry variable
 - overview 225
- DB2_WORKLOAD aggregate registry variable
 - new values 225
- db2adutl command
 - enhancement 60
 - enhancements 63
- db2caem
 - event monitoring 52
- db2cklog command
 - overview 201
- db2ckmig command
 - deprecated 299
- db2dart command
 - enhancements 196
- DB2DETAILDEADLOCK event monitor
 - deprecated 298
- db2diag command
 - enhancements 196
 - parameter additions 202
- db2fmp process
 - privileges customization enhancement 104
- db2fodc command
 - enhancements 193, 194, 196
- DB2FODC registry variable
 - enhancements 196
- db2haicu utility (DB2 High Availability Instance Configuration Utility)
 - Solaris SPARC support 58
- db2has
 - deprecated functionality 289
- db2History APIs
 - COBOL and FORTRAN languages
 - deprecated support 297
- db2ilist command
 - options deprecated 311
- db2imigr command
 - deprecated 299
- db2iprune command
 - enhancement 188
- db2iupdt command
 - option deprecated 299
- db2look command
 - DDL statement generation enhancement 19
- db2mtrk command
 - SYSMON authority added 104
- db2pd
 - history
 - fenced routine history 203

- db2pd command
 - enhancements 193, 195, 196
 - keywords added 49
- db2relocatedb command
 - enhancements 18
- DB2RESILIENCE environment variable
 - overview 225
- db2rfpen command
 - option deprecated 296
- db2rspgn command
 - Linux support added 184
 - UNIX support added 184
- DB2SE_USA_GEOCODER
 - deprecated functionality 294
- db2secv82 command
 - discontinued 310
- db2snapcore
 - new script 196
- db2support command
 - enhancements 193, 196
 - new options 203
- db2trc command
 - application handle 201
 - application id 201
 - enhancements 196
- db2trcoff
 - new script 196
- db2trcon
 - new script 196
- db2uiddl command
 - discontinued 309
- db2val command
 - overview 186
- DB2WebServices
 - discontinued 309
- DBADM (database administration) authority
 - changes 98, 246
- dbheap database configuration parameter
 - change 238
- dec_to_char_fmt database configuration parameter
 - overview 238
- declared temporary tables
 - XML data
 - overview 24
- DECOMPOSE XML DOCUMENTS command
 - overview 32
- deprecated functionality
 - APIs
 - summary 292
 - commands
 - db2ckmig 299
 - db2has 289
 - db2imigr 299
 - LIST TABLESPACE CONTAINERS 292
 - LIST TABLESPACES 292
 - MIGRATE DATABASE 299
 - DB2 administration server (DAS) 286
 - DB2 Health Advisor 289
 - DB2SE_USA_GEOCODER 294
 - health indicators 288
 - health monitor 288
 - Microsoft Systems Management Server 301
 - monitoring routines and views 302
 - statistics event monitor
 - metrics reporting in details_xml 303
 - summary 219, 275, 313
 - Systems Management Server 301

- deprecated functionality (*continued*)
 - Visual Studio 2005 support 290
- DESCRIBE command
 - enhancement 17, 29
 - output changes 233
- details_xml
 - deprecation in the statistics event monitor 303
- diagnostic information
 - alternative path enhancement 200
- diagnostic logs
 - size control 60, 204
- diagpath configuration parameter
 - enhancements 202
- diagsize database manager configuration parameter
 - overview 222
- discontinued functionality
 - commands
 - db2secv82 310
 - db2uiddl 309
 - GET AUTHORIZATIONS 310
 - DB2 Geodetic Data Management Feature 290
 - IBM DB2 Everyplace 309
 - sqluadau API 311
 - summary 219, 305, 313
- distribution maps
 - size increased 15
- documentation
 - overview 345
 - PDF files 345
 - printed 345
 - terms and conditions of use 354
- DOUBLE scalar function
 - return behavior changed 266
- dyn_query_mgmt configuration parameter
 - deprecated 238

E

- embedded processes
 - SAS 109
- encryption
 - enhancement 100
- enhancement 190, 217
- environment variables
 - changes 225
- Event Analyzer
 - deprecated 277
- event monitoring
 - capture activity event monitoring 52
 - db2caem 52
 - db2support 52
 - dynamic SQL statements in package cache 48
 - static SQL statements in package cache 48
- event monitors
 - statistics
 - XML document for system metrics 56
- EXPLAIN authority
 - overview 98
- external scalar functions
 - OUT and INOUT parameters supported 130

F

- FCM (fast communication manager)
 - configuration parameters
 - fcm_parallelism 78

- first occurrence data capture (FODC)
 - enhancements 194
- fix packs
 - enhancements summary 183
 - space requirements reduced 189
 - summary
 - DB2 Connect 211
 - DB2 for Linux, UNIX, and Windows 327
- FOR UPDATE clause
 - overview 115
- functions
 - additions 254
 - changes 254
 - deprecated
 - list 254
 - LONG_VARCHAR 291
 - LONG_VARGRAPHIC 291
 - enhancement 171
 - table
 - ADMIN_EST_INLINE_LENGTH 31, 75
 - ADMIN_IS_INLINED 31, 75

G

- GB 18030 code set
 - DB2CODEPAGE 191
- GB18030
 - Windows client 191
- General Parallel File System (GPFS)
 - NO FILE SYSTEM CACHING new default 224
- Geodetic Data Management Feature
 - discontinued 290
- Geodetic Extender
 - discontinued 290
- GET AUTHORIZATIONS command
 - discontinued 310
- global registry
 - changed 241
- global variables
 - XML 26
- GSKit 105

H

- HADR
 - setting up automatic management scripts (Windows) 62
- Health Center
 - deprecated 277
- health indicators
 - deprecated 288
- health monitor
 - deprecated 288
- help
 - configuring language 350
 - SQL statements 349
- high availability
 - enhancements summary 57
- High Availability Disaster Recovery (HADR)
 - standby database
 - read operations overview 59
- high water marks
 - free space reclamation overview 11
- high watermarks
 - workload management additions 89
- highlighting conventions xiii

- history file
 - change in locking 235
- HP-UX
 - 32-bit client support
 - deprecated support 297

I

- I/O completion ports (IOCPs)
 - AIO support 76
 - new default 76
- IBM data server clients
 - enhancements 132
 - installing
 - db2dsdriver configuration file 130
 - Sysplex support added 149
- IBM Data Server Driver 131
- IBM Data Server Driver configuration keywords
 - default value changes 252
- IBM Data Server Driver for ODBC and CLI
 - merge module changes 271
- IBM Data Server Driver Package
 - enhancements 148
 - environment variables 131
 - Sysplex support added 149
- IBM data server drivers
 - enhancements 132
 - name changes 3
 - Sysplex support added 149
- IBM Data Server Provider for .NET
 - enhancements 161
- IBM Database Add-Ins for Visual Studio
 - enhancements 126
 - install 188
- IBM Database Enterprise Developer Edition
 - IBM solidDB Universal Cache 190
- IBM DB2 Everyplace
 - discontinued functionality 309
- IBM Global Security Kit 105
- IBM solidDB Universal Cache
 - IBM Database Enterprise Developer Edition 190
- IBM Tivoli System Automation for Multiplatforms (SA MP)
 - support enhanced 188
- ibm_db API
 - overview 123
- IBM_DB Ruby driver and Rails adapter
 - trusted contexts 149
- ibm_db_dbi API
 - overview 123
- ibm_db_sa adaptor
 - overview 123
- Index Coexistence 180
- index compression
 - overview 7
- index over XML data
 - enhancement 33
- index reorganization
 - progress information 49
- indexes
 - data partitions 27, 73
 - partitioned
 - overview 27, 73
- Indoubt Transaction Monitor
 - deprecated 277
- inline storage
 - LOBs
 - enhancements 31, 75

- installation
 - changes summary 236
 - enhancements
 - Linux and UNIX 187
 - UNIX 187
 - enhancements summary 183
 - images
 - enhancement 188
- installFixPack command 190, 217
 - enhancement 189
- installing
 - IBM data server clients
 - db2dsdriver configuration file 130
- instances
 - creating
 - shared system support 184
- integer division changes 272
- INTERACTIVE response file keyword 241
- invalidation
 - soft 112
- isolation levels
 - fullselect clauses support 76
 - subselect clauses support 76

J

- JDBC
 - enhancements summary 132
- Journal
 - deprecated 277

L

- large objects (LOBs)
 - CLI retrieval enhancement 150
 - enhancements 31, 75
 - inline 31, 75
 - storage
 - enhancement 31, 75
- LD_LIBRARY_PATH 105
- LIBPATH 105
- License activation process
 - DB2 Connect
 - System z 189
- License Center
 - deprecated 277
- license policies
 - enforcement changes 237
 - setting
 - changes 237
- licenses
 - changes 237
 - types 190
- Lightweight Directory Access Protocol (LDAP)
 - transparent LDAP overview 103
- LIST DATABASE PARTITION GROUPS command
 - SYSMON authority added 104
- LIST DRDA INDOUBT TRANSACTIONS command
 - SYSMON authority added 104
- LIST PACKAGES command
 - SYSMON authority added 104
- LIST TABLES command
 - SYSMON authority added 104
- LIST TABLESPACE CONTAINERS command
 - deprecated 292
 - SYSMON authority added 104

- LIST TABLESPACES command
 - deprecated 292
 - SYSMON authority added 104
- LIST UTILITIES command
 - SYSMON authority added 104
- locking intent
 - subselect clauses and fullselect clauses support 76
- locklist configuration parameter
 - new range 238
- locks
 - event reporting enhancements 46
- log sequence numbers (LSNs)
 - limit increase 254
- logbufsz database configuration parameter
 - changes 238
- logfilesiz database configuration parameter
 - changed functionality 238
- logical data groups
 - activymetrics 53
- logprimary database configuration parameter
 - changes 238
- logs
 - enhancements summary 57
 - non-buffered I/O default change
 - primary 231
 - secondary 231
- LONG VARCHAR data type
 - deprecated 291
- LONG VARGRAPHIC data type
 - deprecated 291
- LONG_VARCHAR function
 - deprecated 291
- LONG_VARGRAPHIC function
 - deprecated 291

M

- manageability
 - enhancements summary 9
- materialized query tables (MQTs)
 - matching enhancements 74
- maxOccurs attribute
 - parsing changes 234
- MemberConnectTimeout 131
- Memory Visualizer
 - deprecated 277
- merge modules
 - .NET, ODBC, and CLI combined 271
- merged backups
 - description 63
- Microsoft Systems Management Server
 - deprecated 301
- MIGRATE DATABASE command
 - deprecated 299
- MIGRATE_PRIOR_VERSIONS response file keyword 300
- migration
 - deprecated commands 299
- modules
 - overview 169
- mon_act_metrics configuration parameter
 - overview 238
- MON_BP_UTILIZATION view
 - enhancement 235
- mon_deadlock configuration parameter
 - overview 238
- MON_GET_BUFFERPOOL table function
 - enhancement 235

- MON_GET_MEMORY_POOL table function
 - overview 54
- MON_GET_MEMORY_SET table function
 - overview 54
- MON_GET_TABLE table function
 - enhancement 235
- MON_GET_TABLESPACE table function
 - enhancement 193, 235
- mon_locktimeout configuration parameter
 - overview 238
- mon_lockwait configuration parameter
 - overview 238
- mon_lw_thresh configuration parameter
 - overview 238
- mon_obj_metrics configuration parameter
 - overview 238
- mon_req_metrics configuration parameter
 - overview 238
- mon_uow_data configuration parameter
 - overview 238
- monitor elements
 - act_remapped_in
 - overview 92
 - act_remapped_out
 - overview 92
 - num_remaps 92
 - row-based formatting
 - support added 51
 - time-spent 44
- monitoring
 - database configuration parameters added 41
 - dynamic SQL statements in package cache 42
 - enhancements 37, 38, 39
 - event monitors
 - maximum number 39
 - last referenced date 45
 - metrics
 - logical data groups 55
 - monitor elements enhancements 41
 - section explain 47
 - SQL statement access plan support
 - section actuals 48
 - SQL statement access plans 47
 - static SQL statements in package cache 42
 - summary 37
 - workload management enhancements 89, 92
- monitoring interfaces
 - FCM support added 52
 - locking support added 40
- monitoring reports
 - generation 50
- moving data
 - Worksheet Format (WSF) deprecated 291
- Multicultural support
 - enhancements summary 191
- multidimensional clustering (MDC) tables
 - reclaiming extents 13
 - XML support 23

N

- named arguments
 - procedures 121
- Net Search Extender
 - checknseindex command
 - enhancement 131
 - validating text index 131

- Net Search Extender (NSE)
 - authorization changes 249
 - commands
 - authorization changes 249
 - deprecated 294
 - deprecated features 294
 - enhancements summary 179
 - full-text searches
 - partitioned database support extended 179
 - partitioned table support added 179
 - incremental updates 180
- Netscape browser support
 - discontinued 307
- new features
 - summary 1
- non-buffered I/O
 - log file changes 231
- notices 357
- num_remaps monitor element 92
- NUMBER data type
 - overview 83
- number_compat mode 272

O

- object management
 - last referenced date 45
- ODBC
 - merge modules packaging simplified 271
- online table moves
 - ADMIN_MOVE_TABLE procedure
 - overview 14
 - recoverable 64
- Optim tools
 - Control Center comparison 277
 - mapping to Control Center tools 277
- optimization guidelines
 - XML data and XQuery 32
- optimization profiles
 - enhancement 69
- Oracle
 - data dictionary--compatible views 82
 - environment setup 84
- ordering DB2 books 348

P

- package cache event monitor
 - supported 48
- packages
 - dynamic packages binding enhancement 150
- parallelism
 - FCM 78
- parameter markers
 - CLI enhancement 150
- parsing
 - implicit
 - maxOccurs attribute 234
- partitioned database environments
 - cost model improved 70
 - Windows 32-bit support removal 307
 - XML data 25
- partitioned indexes
 - overview 27, 73
- partitioned tables
 - detaching data partitions 233

- partitioned tables (*continued*)
 - detaching partitions 77
 - partitioned indexes
 - new default 221
 - reorganization 17
 - XML data
 - overview 22
- passwords
 - maximum length enhancement 103
- PATH 105
- pckcachesz database configuration parameter
 - new range 238
- performance
 - enhancements
 - summary 67
 - scan sharing 71
- PHP
 - trusted contexts
 - overview 149
- pinging
 - CLI application enhancements 150
- PL/SQL
 - compilation support 82
 - statements
 - BULK COLLECT INTO clause 273
- predicate pushdown query optimization
 - overview 30
- priority aging
 - overview 92
- problem determination
 - enhancements summary 193
 - information available 354
 - tutorials 354
- procedures
 - ADMIN_MOVE_TABLE 14
 - common SQL API
 - overview 125
 - DEFAULT keyword change 267
 - named arguments 121
 - parameters
 - default 121
 - enhancement 121
- processes
 - SAS embedded 109
- proxy nodes
 - Tivoli Storage Manager (TSM)
 - overview 60
- public aliases
 - overview 121
- public synonyms
 - overview 121
- Python
 - extensions added 123

Q

- queries
 - access plan reuse overview 68
- Query Patroller
 - deprecated 286
- Query Patroller Center
 - deprecated 286
- quiesced instance
 - RESTRICTED ACCESS option
 - enhancement 18

R

- reclaimable storage
 - automatic storage table spaces 11
 - DMS table spaces 11
- recovery
 - enhancements summary 57
- REDISTRIBUTE DATABASE PARTITION GROUP command
 - enhancements 19
- registry files
 - removed 241
- registry variables
 - additions 225
 - changes 225
 - deprecated functionality 295
 - discontinued support 312
- relocate database command
 - enhancements 18
- REORG INDEXES command
 - CONVERT option deprecated 306
- REORG-recommended operations
 - XML data support added 30
- replication
 - source tables
 - compression overview 7
- resiliency
 - enhancements summary 57
 - errors and traps detection improvements 58
- resources
 - Optim Performance Manager 78
- response files
 - CONFIG_ONLY keyword deprecated 300
 - INTERACTIVE keyword changes 241
 - keywords
 - addition 185
 - MIGRATE_PRIOR_VERSIONS keyword deprecated 300
 - uninstallation
 - enhancements 185
- restoring
 - transporting database schemas
 - overview 61
- revalidation
 - automatic 112
 - soft 112
- routines
 - additions 254
 - changes 254
- roving high availability (HA) support
 - supported 63
- row data types
 - overview 176
- rows
 - count retrieval
 - enhancement 150

S

- SA MP
 - setting up automatic management scripts (Windows) 62
- samples
 - additions 127
- Satellite Administration Center
 - deprecated 277
- scalability
 - database partition server enhancement 16
- scalar functions
 - CHAR return behavior change 265

- scalar functions (*continued*)
 - DOUBLE return behavior change 266
 - enhancements 116
- scan sharing
 - overview 71
- SDKs
 - version 1.4.2 deprecation 292
- SECADM (security administrator) authority
 - changes 98, 245
- security
 - changes summary 243
 - enhancements summary 97
 - plug-ins
 - LDAP (Lightweight Directory Access Protocol) 103
- security connection parameter 101
- SELECT INTO statement
 - FOR UPDATE clause 115
- SERVER_ENCRYPT authentication type
 - enhancement 100
- service classes
 - tiering 92
- SHLIB_PATH 105
- shredding XML documents
 - enhancement 32
- SNAP_GET_AGENT_MEMORY_POOL
 - deprecated functionality 302
- SNAP_GET_DB_MEMORY_POOL
 - deprecated functionality 302
- SNAP_GET_DBM_MEMORY_POOL
 - deprecated functionality 302
- SNAP_GET_TAB_V91
 - deprecated functionality 302
- SNAPTAB
 - deprecated functionality 302
- soft invalidation
 - overview 112
- spatial data
 - partitioned database environments 78
- SQL
 - administrative routines
 - additions 13
 - administrative views
 - additions 13
 - alternative syntax support 84
 - compatibility enhancements 81, 85
- SQL Procedural Language (SQL PL)
 - data types
 - anchored 173
 - associative array 175
 - Boolean 174
 - cursor 175
 - new 173
 - row 176
 - DEFAULT keyword specification 267
 - enhancements summary 169
 - statements
 - enhancement 171
- SQL statements
 - help
 - displaying 349
 - OR REPLACE clause 110
- SQLADM (SQL administration) authority
 - overview 98
- SQLAlchemy
 - adapter added 123
- SQLCreatePkg API 150
- sqlmgdb API
 - deprecated 299
- sqlmgdb API
 - deprecated 299
- SQLROWSREAD activity threshold
 - overview 91
- sqluadai API discontinued 311
- sqlugrpn API
 - deprecated 293
- sqlugtpi API
 - deprecated 293
- srvcon_auth configuration parameter
 - changes 222
- SSL
 - enhancements 101
 - setup enhancement 247
- ssl_cipherspecs configuration parameter
 - overview 101, 222
- ssl_clnt_keydb configuration parameter
 - overview 222
- ssl_clnt_stash configuration parameter
 - overview 222
- ssl_svcname configuration parameter
 - overview 101, 222
- ssl_svr_keydb configuration parameter
 - overview 101, 222
- ssl_svr_label configuration parameter
 - overview 101, 222
- ssl_svr_stash configuration parameter
 - overview 101, 222
- ssl_versions configuration parameter
 - overview 101, 222
- SSLClientKeystash connection parameter
 - overview 101
- SSLClientKeystoredb connection parameter
 - overview 101
- statement concentrator
 - CLI enhancement 150
 - overview 68
- statistical views
 - RUNSTATS command 69
- statistics
 - collection
 - workload management enhancements 89, 92
- stmt_conc database configuration parameter
 - overview 238
- storage paths
 - automatic
 - dropping 10
- stored procedures
 - result changes 268
- strong encryption
 - enhancement 100
- super asynchronous (SUPERASYNC) mode
 - overview 64
- synchronization modes
 - SUPERASYNC added 64
- synonyms
 - public 121
- SYSADM (system administration) authority
 - changes 98, 243
- SYSCAT views
 - additions 254
 - changes 254
- SYSMON (system monitor) authority
 - db2mtrk command added 104
 - LIST commands added 104

- Sysplex
 - IBM data server client support added 149
- system catalogs
 - views
 - additions 254
 - changes 254
- system information
 - table functions 54
- System z
 - DB2 Connect
 - license activation process 189
- system-defined modules
 - overview 124
- Systems Management Server
 - deprecated 301

T

- table functions
 - deprecated functionality
 - summary 254
- table reorganization
 - progress information 49
- table space states
 - modification status enhancement 193
- table spaces
 - increased capacity limit 15
 - rebalancing
 - overview 10
- tables
 - moving online
 - overview 14
 - partitioned
 - partitioned index overview 27, 73
- Task Center
 - deprecated 277
- temporary tables
 - compression 6
 - LOB data support added 120
- terms and conditions
 - publications 354
- thresholds
 - AGGSQLTEMPSPACE
 - overview 91
 - CPUTIME
 - overview 91
 - SQLROWSREAD
 - overview 91
- TIMESTAMP data type
 - enhancements 120
- Tivoli Storage Manager (TSM)
 - db2adutl command enhancement 63
- tools 302
- trace command
 - application handle 201
 - application id 201
- trace facility
 - improvements 194
- transaction control
 - CLI rollback enhancement 150
- transaction monitoring 43
- transactions
 - ALTER TABLE operations increased limit 111
 - autonomous 123
- Transport Layer Security (TLS)
 - enhancements 101

- transports
 - database schemas
 - overview 61
- triggers
 - SQL PL enhancements 171
 - SQL PL statements 171
- troubleshooting 302
 - enhancements summary 193
 - large systems
 - enhancements 196
 - load operation enhancement 195
 - log files
 - enhancement 201
 - online information 354
 - tutorials 354
- TRUNCATE statement
 - overview 115
- trusted contexts
 - .NET support added 149
 - IBM_DB Ruby driver support
 - overview 149
 - PHP extensions support added 149
- tutorials
 - list 354
 - problem determination 354
 - troubleshooting 354
 - Visual Explain 354
- type-1 indexes
 - discontinued
 - details 306

U

- unary operators
 - changed return data types 267
- uninstalling
 - response file support enhancements 185
- unit of work event monitor
 - overview 43
- universal fix packs
 - Windows support added 189
- untyped NULL keyword specification
 - changes 264
- update service
 - enabled by default 186
- updates
 - DB2 Information Center 351, 352
- upgrades
 - applications
 - deprecated commands 299
 - clients
 - deprecated commands 299
 - DB2 servers
 - deprecated commands 299
 - enhancements summary 183
 - response file keyword additions 185
 - routines
 - deprecated commands 299
- user-defined functions (UDFs)
 - overridden by SYSIBM functions 263
 - XML data type support added 24

V

- validation
 - DB2 installation 186

- VARCHAR2 data type
 - overview 83
- views
 - additions 254
 - changes 254
 - compatible with Oracle data dictionary 82
- Visual Studio 2005
 - deprecated support 290

W

- Web Object Runtime Framework (WORF) support
 - discontinued 308
- WLMADM (workload administration) authority
 - overview 98
- workload management
 - monitoring 39
- Worksheet Format (WSF)
 - deprecated 291

X

- XML
 - compressing documents 5, 35
 - enhancements
 - compressing documents 5, 35
 - decomposition 32
 - indexing 33
 - summary 21
 - global variables 26
 - multidimensional clustering (MDC) tables 23
 - partitioned database environments 25
 - processing improvement 30
 - stored procedure result changes 268
- XML data
 - compressing 5, 35
 - distribution statistics 34
 - querying XML data 34
- XML data type
 - user-defined functions (UDFs) 24
- XML decomposition
 - enhancements 32
- XML documents
 - compressing 5, 35
 - storage
 - enhancement 31, 75
 - type annotation changes 270
- XML Extender
 - discontinued 308
- XML function parameters
 - compiled SQL functions 26
- XML indexes
 - concurrency enhancement 33
- XQuery
 - optimization guidelines 32



Printed in USA

SC27-2463-03



Spine information:

DB2 for Linux, UNIX, and Windows **Version 9 Release 7**

What's New for DB2 Version 9.7

