

IBM DB2 10.1
for Linux, UNIX, and Windows

What's New for DB2 Version 10.1

IBM

IBM DB2 10.1
for Linux, UNIX, and Windows

What's New for DB2 Version 10.1

IBM

Note

Before using this information and the product it supports, read the general information under Appendix D, "Notices," on page 211.

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 <http://www.ibm.com/shop/publications/order>
- To find your local IBM representative, go to the IBM Directory of Worldwide Contacts at <http://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	ix
Who should use this book	ix
How this book is structured	ix
Highlighting conventions	xi
<hr/>	
Part 1. What's New	1
Chapter 1. Highlights of DB2 Version 10.1	3
Chapter 2. Product packaging enhancements	7
Chapter 3. Manageability enhancements	9
Row compression rates and ease of use have improved	9
Multi-temperature data storage can provide fast access to data	10
Storage management has been improved	11
Table spaces inherit media attributes from storage groups	12
New mechanism to reclaim unused space in indexes	13
New insert time clustering tables	14
The db2move command now supports parallel processing.	14
Partitioned tables remain accessible to queries when a data partition is added or attached	14
Chapter 4. pureXML enhancements	17
New types supported for indexes over XML	17
Functional XML indexes can speed up queries.	17
New binary XML format improves performance for certain Java clients	19
XML casting and truncation errors reduced.	20
Performance improved for certain XML queries	22
Chapter 5. Monitoring enhancements	25
New event monitor tracks configuration and registry changes and DDL and utility executions	25
New usage list object identifies statements that affect tables or indexes	26
New STATEMENT domain for thresholds lets you define thresholds for statements that contain specific text.	26
New and changed functions for accessing monitoring information	27
Executable ID list is included in unit of work event monitor information.	31
All event monitors now support the WRITE TO TABLE target	31
Existing event monitors that write to tables can be altered to capture additional logical data groups.	32
Event monitor tables can now be upgraded.	33
Pruning data from unformatted event tables is now supported	33
New monitor elements provide added insight into the operation of your DB2 server	34
Chapter 6. High availability, backup, logging, resiliency, and recovery enhancements	47
HADR now supports multiple standby databases.	47
High-speed ingest utility allows movement of real-time data without affecting availability	48
Storing archived log files requires less disk space.	48
HADR delayed replay provides protection against application errors	49
HADR log spooling prevents spikes on throughput	49
Replication improvements	50
Chapter 7. Performance enhancements	51
Enhanced query performance for common SQL statements	51
RUNSTATS and database statistics improvements	53
Optimization profile supports registry variables and inexact matching.	54
Statistical views enhance statistics and statistic gathering for the query optimizer	54

Intra-partition parallelism improvements	55
Enhanced memory sharing on large POWER7 systems running AIX	57
Improved query performance through more efficient data and index prefetching	57
Improved performance for queries on tables with composite indexes	58
Performance of star schema based queries is improved	60
Chapter 8. SQL compatibility enhancements	63
Expanded support for triggers	63
Declared types and procedures	64
New scalar functions	64
Chapter 9. Workload management enhancements	65
DB2 WLM dispatcher manages service class CPU resource allocations	65
DB2 WLM can prioritize activities based on the accessed data	68
DB2 workload manager is now available in the DB2 pureScale environment.	69
Chapter 10. Security enhancements	71
Row and column access control (RCAC) enhances data security	71
Chapter 11. Application development enhancements	73
Built-in global variables expand SQL programming capabilities	73
Use Java generic table functions for custom analytics	74
Manage and query time-based data using temporal tables	75
RDF application development support has been added	75
IBM data server clients and drivers enhancements	76
JDBC and SQLJ support has been enhanced	76
Chapter 12. DB2 Text Search enhancements	79
DB2 Text Search improvements for search	79
DB2 Text Search stand-alone server deployment	79
DB2 Text Search supports partitioned database environments.	80
DB2 Text Search supports partitioned tables	80
Chapter 13. Installation and upgrade enhancements	81
Installation commands have been enhanced	81
New response file keywords have been added.	83
New db2prereqcheck command checks prerequisites before starting an installation	83
DB2 License compliance report enhanced	83
DB2 pureScale Feature included in DB2 server edition installations.	84
DB2 Spatial Extender is now included in the DB2 database product media	84
Installation of IBM Data Studio integrated into the DB2 installation process	84
Chapter 14. DB2 pureScale Feature enhancements.	87
DB2 pureScale Feature included in DB2 server edition installations.	87
DB2 pureScale Feature support added for AIX servers on RoCE networks	88
Installation commands have been enhanced	88
Range partitioning is now available for DB2 pureScale environments	90
DB2 pureScale Feature can be validated using db2val command.	90
db2cluster command now supports repairing an instance domain and controlling when an automatic failback occurs	91
New CURRENT MEMBER default value improves DB2 pureScale performance	92
New monitoring routine provides insight into group buffer pool usage	93
DB2 workload manager is now available in the DB2 pureScale environment.	93
Diagnostic data collection for DB2 pureScale environments has been improved.	93
Chapter 15. Multicultural support enhancements.	95
New locale-sensitive UCA-based collations based on CLDR 1.8.1	95
New locales based on CLDR 1.8.1	96
Additional territory codes supported by IBM data server clients.	96

Part 2. What's changed	97
Chapter 16. Administration changes summary.	99
Replication Center is now a stand-alone tool	99
Collection of package list information has changed	99
DB2 Text Search security model changes	100
DB2 Text Search index location has changed	100
DB2 Text Search scheduler changes	101
DB2 Text Search administrative commands and stored procedures have changed	101
Names of TOTALDBPARTITIONCONNECTIONS and TOTALSCPARTITIONCONNECTIONS thresholds have been changed	102
Section actuals now include object statistics	103
The default diagnostic log paths are now split by default.	103
Some database manager configuration parameters have been changed	105
Some registry and environment variables have changed	107
Chapter 17. Database setup and product installation changes summary	111
Informational constraints can now be TRUSTED or NOT TRUSTED	111
IBM Tivoli System Automation for Multiplatforms (SA MP) is now automatically installed	112
Some database configuration parameters have been changed	112
Chapter 18. Security changes summary	117
Kerberos authentication changes (UNIX)	117
Chapter 19. Application development changes summary.	119
XML casting error (SQL16061N) less likely to occur.	119
Optimizer can now choose VARCHAR indexes for queries that contain fn:starts-with	120
CREATE GLOBAL TEMPORARY TABLE and DECLARE GLOBAL TEMPORARY TABLE statements have been changed	121
Some catalog views and built-in routines changed to include member information	121
Some built-in string functions have changed result data types in Unicode databases.	124
Some special registers have changed	125
Some system catalog views, built-in functions and global variables, built-in administrative routines and views have been added and changed	125
Chapter 20. DB2 command and SQL statement changes summary	131
db2cat command output text has changed.	131
db2ckupgrade command enhanced for partitioned environments	132
Some parameters for commands that manage instances have changed	132
db2evtbl command has been changed	133
db2exfmt command output has changed for partitioned tables	134
db2pd command has changed to support new functionality	134
ALTER TABLE and CREATE TABLE statement have been changed	135
ALTER TABLESPACE statement has a new clause	136
ALTER WORKLOAD and CREATE WORKLOAD statements have changed	137
DETAILED keyword in CREATE INDEX statement now indicates a changed default behavior	138
Chapter 21. Deprecated functionality	139
Activity monitor routines have been deprecated.	140
Agent priority of service classes has been deprecated	141
Collations based on the Unicode Collation Algorithm of the Unicode Standard version 4.0.0 have been deprecated	141
IBM Tivoli System Automation for Multiplatforms (SA MP) is now automatically installed	142
SMS permanent table spaces have been deprecated.	142
Automatic statistics profiling is deprecated	143
Some version-suffixed SQL administrative routines have been deprecated	144
Net Search Extender has been deprecated	145
db2IdentifyType1 command has been deprecated	145
db2_install command has been deprecated (Linux and UNIX)	146

dynexpln command has been deprecated	146
PRUNE LOGFILE command has been deprecated	147
Some parameters of the CREATE DATABASE command have been deprecated	148
Some parameters for commands that manage instances have changed	148
-flushbp parameter of the db2pdcfg command has been deprecated	149
REORG INDEXES/TABLE command parameters and parameter values for related DB2 API data structures have been deprecated or discontinued	150
ALTER DATABASE statement has been deprecated.	151
Some registry and environment variables have been deprecated	151
Some database configuration parameters have been deprecated or discontinued	152
Some monitoring interfaces for HADR have been deprecated	153
The -global option for troubleshooting tools has been deprecated	154

Chapter 22. Discontinued functionality 155

32-bit client support has been discontinued (HP-UX)	156
DB2 JDBC Type 2 Driver has been discontinued	156
DB2SE_USA_GEOCODER has been discontinued	157
Distributed installation support with Microsoft Systems Management Server has been discontinued (Windows)	157
Some operating systems are no longer supported	158
Some version-suffixed SQL administrative routines have been discontinued	158
Worksheet Format (WSF) for Import and Export utilities have been discontinued.	159
Microsoft Visual Studio 2005 support has been discontinued	159
Query Patroller has been discontinued	159
Control Center tools have been discontinued	160
IBM DB2 Geodetic Data Management Feature has been discontinued.	163
Support in COBOL, FORTRAN, and REXX for DB2 APIs that manage database history records has been discontinued	163
db2imigr and db2ckmig commands have been discontinued.	164
-file parameter of the db2flsn and db2rfpen commands has been discontinued	164
-s parameter of the db2iupdt command has been discontinued	165
Some registry and environment variables have been discontinued.	165
Some database configuration parameters have been deprecated or discontinued	168

Chapter 23. Summary of deprecated and discontinued DB2 functionality in Version 10.1 and earlier releases 171

Part 3. DB2 Connect enhancements and changes summary 189

Chapter 24. DB2 Version 10.1 enhancements and changes that affect DB2 Connect 191

Part 4. Appendixes 193

Appendix A. Functionality in DB2 features and DB2 product editions 195

Appendix B. Functionality in DB2 features in DB2 Connect product editions 199

Appendix C. Overview of the DB2 technical information 201

DB2 technical library in hardcopy or PDF format	201
Displaying SQL state help from the command line processor	204
Accessing different versions of the DB2 Information Center	204
Updating the DB2 Information Center installed on your computer or intranet server	204
Manually updating the DB2 Information Center installed on your computer or intranet server	206
DB2 tutorials	207
DB2 troubleshooting information	208
Terms and conditions.	208

Appendix D. Notices 211

Index 215

About this book

This book provides information about the new and changed functionality included in the Version 10.1 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 10.1 for Linux, UNIX, and Windows and in DB2 Connect Version 10.1 and what differences exist between Version 10.1 and Version 9.7 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 10.1, read Part 1, “What’s New,” on page 1.

For information about the changed, deprecated, or discontinued functionality in Version 10.1, read Part 2, “What’s changed,” on page 97. This information points out important changes that you need to know before using Version 10.1.

For information about DB2 Connect, read Part 3, “DB2 Connect enhancements and changes summary,” on page 189.

How this book is structured

The following topics are covered:

Part 1: What’s New

Chapter 1, “Highlights of DB2 Version 10.1,” on page 3

This chapter describes the product highlights about important new features and enhancements.

Chapter 2, “Product packaging enhancements,” on page 7

This chapter describes the product packaging changes introduced in Version 10.1.

Chapter 3, “Manageability enhancements,” on page 9

This chapter describes the new features and enhancements that help you spend less time managing your databases.

Chapter 4, “pureXML enhancements,” on page 17

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

Chapter 5, “Monitoring enhancements,” on page 25

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 47

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 51

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 63

This chapter describes the new features and enhancements that help you port existing database applications from other vendors to DB2 Version 10.1 environments.

Chapter 9, “Workload management enhancements,” on page 65

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 71

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

Chapter 11, “Application development enhancements,” on page 73

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

Chapter 12, “DB2 Text Search enhancements,” on page 79

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

Chapter 13, “Installation and upgrade enhancements,” on page 81

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

Chapter 14, “DB2 pureScale Feature enhancements,” on page 87

This chapter describes the new features and enhancements available for support of the DB2 pureScale®.

Chapter 15, “Multicultural support enhancements,” on page 95

This chapter describes the new features and enhancements that make it easier to work with data and database applications that work with multiple national languages.

Part 2: What's changed

Chapter 16, “Administration changes summary,” on page 99

This chapter describes the changes in the existing DB2 functionality related to database administration.

Chapter 17, “Database setup and product installation changes summary,” on page 111

This chapter describes the changes in the existing DB2 functionality related to database setup and product installation.

Chapter 18, “Security changes summary,” on page 117

This chapter describes the changes in the existing DB2 functionality related to security.

Chapter 19, “Application development changes summary,” on page 119

This chapter describes the changes in the existing DB2 functionality related to application development.

Chapter 20, “DB2 command and SQL statement changes summary,” on page 131

This chapter describes the changes to DB2 CLP commands, DB2 system commands, and SQL statements to support new capabilities.

Chapter 21, “Deprecated functionality,” on page 139

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 22, “Discontinued functionality,” on page 155

This chapter lists features and functionality that are unsupported in Version 10.1.

Chapter 23, “Summary of deprecated and discontinued DB2 functionality in Version 10.1 and earlier releases,” on page 171

This chapter lists features and functionality that have been deprecated or discontinued in DB2 Version 10.1.

Part 3: DB2 Connect enhancements and changes summary

Chapter 24, “DB2 Version 10.1 enhancements and changes that affect DB2 Connect,” on page 191

This chapter describes the enhancements, changed functionality, deprecated functionality, and discontinued functionality in Version 10.1 that affect the functionality and capabilities of DB2 Connect.

Part 4: Appendixes

Appendix A, “Functionality in DB2 features and DB2 product editions,” on page 195

This appendix contains information about functionality that is available in DB2 database product editions and DB2 features.

Appendix B, “Functionality in DB2 features in DB2 Connect product editions,” on page 199

This appendix contains information about functionality that is available in DB2 Connect product editions and DB2 features.

Appendix C, “Overview of the DB2 technical information,” on page 201

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

Appendix D, “Notices,” on page 211

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 "FP x " 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. What's New

What's New includes information about new functionality included in Version 10.1.

DB2 Version 10.1 for Linux, UNIX, and Windows delivers new capabilities that can help you manage costs and simplify application development.

Chapter 1, “Highlights of DB2 Version 10.1,” on page 3

This chapter describes the product highlights about important new features and enhancements.

Chapter 2, “Product packaging enhancements,” on page 7

This chapter describes the product packaging changes introduced in Version 10.1.

Chapter 3, “Manageability enhancements,” on page 9

This chapter describes the new features and enhancements that help you spend less time managing your databases.

Chapter 4, “pureXML enhancements,” on page 17

This chapter describes the new pureXML features and enhancements.

Chapter 5, “Monitoring enhancements,” on page 25

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 47

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 51

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 63

This chapter describes the new features and enhancements that help you port existing database applications from other vendors to DB2Version 10.1 environments.

Chapter 9, “Workload management enhancements,” on page 65

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 71

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

Chapter 11, “Application development enhancements,” on page 73

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

Chapter 12, “DB2 Text Search enhancements,” on page 79

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

Chapter 13, “Installation and upgrade enhancements,” on page 81

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

Chapter 14, “DB2 pureScale Feature enhancements,” on page 87

This chapter describes the new features and enhancements available for support of the DB2 pureScale.

Chapter 15, “Multicultural support enhancements,” on page 95

This chapter describes the new features and enhancements that make it easier to work with data and database applications that work with multiple national languages.

Chapter 1. Highlights of DB2 Version 10.1

IBM® DB2 Version 10.1 for Linux, UNIX, and Windows continues to deliver database efficiency, simplicity, and reliability. Important new features and enhancements address the needs of your business. Whether your needs are improving business critical reliability and performance, simplifying consolidation and security, increasing business insight, reducing costs, or providing a resilient system for the valuable information assets of your company, DB2 Version 10.1 delivers.

Adaptive compression

Your DB2 database provides comprehensive compression for almost every type of database object including table data, indexes, temporary tables, XML documents, log files and backup images. In previous versions, classic row compression helped lower storage costs and improve query performance. DB2 Version 10.1 further enhances classical table compression through a new type of compression, adaptive compression.

The adaptive compression feature improves table compression through an advanced row compression technique that uses two levels of compression dictionaries (table-level and page-level) to improve compression ratios, particularly as data changes. Page-level compression dictionaries are smaller than table-level dictionaries so as data changes on a page it is easy to update automatically and quickly, and removes the need to perform table reorganizations to update page-level dictionaries.

Adaptive compression helps you:

- achieve high compression ratios without taking your data offline to perform table reorganizations
- continue to improve query performance
- increase storage savings and system availability
- save money through decreased storage requirements.

Even easier, in DB2 Version 10.1, new tables use adaptive compression by default. Existing tables from previous DB2 releases can easily use adaptive compression by simply enabling it.

For more information about adaptive compression, see “Row compression rates and ease of use have improved” on page 9.

Data security enhancements

DB2 Version 10.1 provides critical enhancements to security and auditing by introducing row and column access control (RCAC) as a solution to help you further secure your data. RCAC is sometimes referred to as fine-grained access control or FGAC.

RCAC security allows you to easily create varying security rules at the data level. These security rules ensure that users, who are members of the approved roles or groups, see only the data they are allowed to see, and removes security constraints

and performance headaches resulting from complex views and predicates. Setup is quick and simple, and security is easy to handle even for complex enterprise systems.

Benefits provided by RCAC include:

- a centralized, enforceable, and auditable process that controls access to data
- lower cost associated with the development and management of access control rules on business sensitive data.
- reduce time to value for business process applications that have compliance or audit requirements.

Even easier, application changes are not required when implementing RCAC.

For more information about RCAC, see “Row and column access control (RCAC) enhances data security” on page 71.

IBM DB2 pureScale Feature enhancements

Reduce risk and cost associated with growing your distributed database solution by providing extreme capacity and application transparency. Designed for continuous availability, high availability capable of exceeding even the strictest industry standard, the IBM DB2 pureScale Feature tolerates both planned maintenance and component failure with ease.

The DB2 pureScale Feature was first introduced in Version 9.8. Version 10.1 builds on DB2 pureScale Feature support.

For more information, see Chapter 14, “DB2 pureScale Feature enhancements,” on page 87.

Multi-temperature storage

Assign priority to data (hot, warm, cold) and dynamically assign it to different classes of storage. For example, transaction records for the current quarter can be stored on high-performance storage, and after the quarter ends and the data is no longer considered hot, it can be moved to cheaper storage. Total cost of ownership savings are considerable, allowing efficient deployment of storage hardware, and minimal administrative overhead.

For more information about multi-temperature storage, see “Multi-temperature data storage can provide fast access to data” on page 10.

Performance improvements

Building on prior release performance improvements (such as, automatic performance improvements and **RUNSTATS** command improvements), DB2 Version 10.1 performance improvements focus on reducing CPU processing time without causing significant administration or application changes. Most performance improvements are implemented by simply upgrading to Version 10.1. You can gain significant performance improvements from improved query optimizer techniques and functionality including star schema query optimization, improved data and index prefetching, and improved use of statistical views. There are also additional **RUNSTATS** command improvements, improved performance for queries on tables with composite indexes, as well as improved multi-core parallelism.

DB2 Version 10.1 makes it simpler for you to write and run efficient SQL queries, as well as, making your existing SQL queries run faster, often without any changes.

For more information, see Chapter 7, “Performance enhancements,” on page 51.

SQL compatibility

If you work with relational database products other than DB2 products, Version 10.1 builds on existing functionality, interfaces, and compatibility features to provide additional enhancements that make DB2 products more familiar to you. These enhancements reduce the time and complexity of enabling applications written for other relational database products to run quickly in a DB2 environment.

For more information, see Chapter 8, “SQL compatibility enhancements,” on page 63.

Time Travel Query

Time Travel Query makes your database time-aware and keeps a history of your data changes using temporal tables. You can travel to the past and query your data as it appeared at different points in time.

Time Travel Query helps you:

- easily make your existing DB2 tables time-aware
- provide a cost-effective means to resolve auditing and compliance issues
- reduce cost with efficient SQL coding of complex time-focused operations to implement and maintain time-aware applications
- reduce application development time by allowing DBAs to use an existing SQL application and run it across different time periods
- reduce your time to deployment through an inexpensive and easy to maintain time-focused data support infrastructure
- create a time-based warehouse at low cost without additional application logic.

Time Travel Query is easily enabled for existing tables using the ALTER TABLE statement, including existing tables from previous DB2 releases. You can store and retrieve time-based data without additional application logic.

For more information about Time Travel Query, see “Manage and query time-based data using temporal tables” on page 75.

DB2 Version 10.1 contains many other enhancements and new functionality. For details, see the enhancements listed in Part 1, “What’s New,” on page 1.

Chapter 2. 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.

IBM has updated product packaging to provide simplicity through fewer packages, and greater value through more function and features included in the base DB2 editions.

To read about these products and to view the related licensing and marketing information, see the DB2 product page at <http://www.ibm.com/software/data/db2/linux-unix-windows>.

Note: In Version 10.1, occurrences of “DB2 pureCluster Feature” in documentation or messages refer to the IBM DB2 pureScale Feature.

Related concepts:

Chapter 14, “DB2 pureScale Feature enhancements,” on page 87

Related reference:

Appendix A, “Functionality in DB2 features and DB2 product editions,” on page 195

“Functionality in DB2 features in DB2 Connect product editions” in DB2 Connect User's Guide

Chapter 3. Manageability enhancements

Version 10.1 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.

The following enhancements are also included:

- Improved row compression rates with greater ease of use (see “Row compression rates and ease of use have improved”)
- Multi-temperature data storage can provide fast access to data (see “Multi-temperature data storage can provide fast access to data” on page 10)
- Management of storage groups has been improved (see “Storage management has been improved” on page 11)
- Table spaces inherit media attributes from storage groups (see “Table spaces inherit media attributes from storage groups” on page 12)
- Introduction of temporal tables as a way of associating time-based state information with your data (see “Manage and query time-based data using temporal tables” on page 75)
- New mechanism to reclaim unused space on indexes for tables that reside in DMS table spaces (see “New mechanism to reclaim unused space in indexes” on page 13)
- Introduction of insert time clustering tables as a way of maintaining data clustering (see “New insert time clustering tables” on page 14)
- The db2move command enhanced (see “The db2move command now supports parallel processing” on page 14)
- Improved accessibility to partitioned tables when adding or attaching new partitions (see “Partitioned tables remain accessible to queries when a data partition is added or attached” on page 14)

Row compression rates and ease of use have improved

DB2 Version 10.1 features some major enhancements to row compression, including improved compression rates, improved ease of use, and higher data availability.

Adaptive compression

In DB2 Version 10.1, table data can be compressed with *page-level compression dictionaries* in addition to the table-level compression dictionary used in earlier versions of the product. In this compression scheme, each page of table data has a page-level compression dictionary that takes into account all of the data that exists within the page. Page-level compression dictionaries are automatically maintained; when there are changes to data in a page, the compression dictionary is dynamically updated. This means that you do not need to perform a table reorganization to compress data on that page. Thus, in addition to improved compression rates, this approach to compression can improve the availability of your data. This method of compressing data rows with both table-level and the automatically updated page-level compression dictionaries is known as *adaptive compression*.

Enabling adaptive compression

You can enable adaptive compression by using the COMPRESS YES ADAPTIVE clause in the CREATE TABLE and ALTER TABLE statements.

Note: The ADAPTIVE keyword is now the default option for the COMPRESS YES clause.

For a table enabled to use adaptive compression, rows are compressed during the following data change operations:

- Inserts
- Updates
- Imports
- Loads
- Redistributes
- Reorgs
- Online table moves

After adaptive compression is enabled for a table with existing data, rows that are subsequently inserted might trigger the creation of additional page-level dictionaries as new data is added.

Related concepts:

"Adaptive compression" in Database Administration Concepts and Configuration Reference

Related reference:

"ALTER TABLE " in SQL Reference Volume 2

"CREATE TABLE " in SQL Reference Volume 2

Multi-temperature data storage can provide fast access to data

You can manage your IT budget more efficiently by configuring your database so that only frequently accessed data (*hot data*) is stored on expensive fast storage, such as solid-state drives (SSD), and infrequently accessed data (*cold data*) is stored on slower, less-expensive storage, such as low-rpm hard disk drives.

As hot data cools down and is accessed less frequently, you can dynamically move it to the slower storage, thereby extending the life of your less expensive storage assets that are used for storing warm and cold data.

In database systems, there is a strong tendency for a relatively small proportion of data to be hot data and the majority of the data to be cold data. Current data is often considered to be hot data, but it typically becomes cold as it ages. These sets of *multi-temperature data* pose considerable challenges to DBAs who want to optimize the use of fast storage by trying not to store cold data there. As a data warehouse consumes more storage, optimizing the use of fast storage becomes increasingly important for managing storage costs.

With your hot data stored on your fastest storage assets, multi-temperature data storage can help to reduce the time it takes to retrieve your most frequently accessed data, while reducing the cost of storing infrequently accessed warm and cold data.

Related concepts:

"DB2 WLM can prioritize activities based on the accessed data" on page 68

"Storage management has been improved"

"Data management using multi-temperature storage" in Database Administration Concepts and Configuration Reference

Related reference:

"ALTER TABLESPACE " in SQL Reference Volume 2

"CREATE TABLESPACE " in SQL Reference Volume 2

"RENAME STOGROUP " in SQL Reference Volume 2

"ALTER STOGROUP " in SQL Reference Volume 2

"CREATE STOGROUP " in SQL Reference Volume 2

Storage management has been improved

A major advance in DB2 Version 10.1 is the ability to create *storage groups*, which are groups of storage paths. A storage group contains storage paths with similar characteristics. Some critical attributes of the underlying storage to consider when creating or altering a storage group are available storage capacity, latency, data transfer rates, and the degree of RAID protection.

These storage groups can be used to create different classes of storage (multi-temperature storage classes) where frequently accessed (or hot) data is stored in storage paths residing on fast storage, while infrequently accessed (or cold) data is stored in storage paths residing on slower, less expensive storage.

After you create storage groups that map to the different classes of storage in your database management system, you can assign automatic storage table spaces to those storage groups, based on which table spaces have hot or cold data. You can use storage groups to physically partition table spaces managed by automatic storage. You can dynamically reassign a table space to a different storage group by using the ALTER TABLESPACE statement with the USING STOGROUP option.

A database-managed table space can be converted to an automatic storage table space by executing an ALTER TABLESPACE statement and specifying the MANAGED BY AUTOMATIC STORAGE clause on the table space. Note that after this is done, it is necessary to perform a rebalance operation on the table space by executing an ALTER TABLESPACE statement and specifying the REBALANCE clause on the table space. In Version 10.1, the rebalance operation is enhanced to manually SUSPEND and RESUME a rebalance operation during performance sensitive periods.

You can take further advantage of organizing your data into storage groups by configuring DB2 workload manager (WLM) to prioritize activities based on the priority of the data being accessed.

The ADMIN_GET_STORAGE_PATHS table function can be used to get the list of automatic storage paths for each database storage group, including file system information for each storage path. Other table functions that have been added or modified to support the monitoring of storage groups include: MON_GET_REBALANCE_STATUS, MON_GET_TABLESPACE and MON_GET_CONTAINER.

To support the management of storage groups, the following SQL statements and DB2 commands have been added or modified:

- The ALTER STOGROUP statement is new.
- The CREATE STOGROUP statement is new.
- The RENAME STOGROUP statement is new.
- The COMMENT statement has the new STOGROUP clause.
- The DROP statement has the new STOGROUP clause.
- The ALTER TABLESPACE statement has the new USING STOGROUP clause.
- The CREATE TABLESPACE statement has the new USING STOGROUP clause.
- The **db2pd** command has the new **-storagegroups** parameter.
- The **-l** parameter of the **db2look** command has been modified to generate DDL statements for user-defined storage groups.

To support redirected restore of storage groups, the following commands and APIs have been added or modified:

- The **RESTORE DATABASE** command has a new **-USING STOGROUP storagegroup-name-** parameter.
- The **SET STOGROUP PATHS** command is new.
- The db2Restore API has a new **piStogroup** parameter in the db2RestoreStruct data structure.
- The db2SetStogroupPaths API is new.

Related concepts:

"Multi-temperature data storage can provide fast access to data" on page 10 "Storage groups" in Database Administration Concepts and Configuration Reference

Related reference:

"ALTER TABLESPACE " in SQL Reference Volume 2

"CREATE TABLESPACE " in SQL Reference Volume 2

"DROP " in SQL Reference Volume 2

"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

"ADMIN_GET_STORAGE_PATHS table function - retrieve automatic storage path information" in Administrative Routines and Views

"MON_GET_REBALANCE_STATUS table function - get rebalance progress for a table space" in Administrative Routines and Views

"RENAME STOGROUP " in SQL Reference Volume 2

"ALTER STOGROUP " in SQL Reference Volume 2

"CREATE STOGROUP " in SQL Reference Volume 2

Table spaces inherit media attributes from storage groups

When you use the CREATE TABLESPACE statement to create a table space, you can specify that the table space dynamically inherits media attributes from its associated storage group. You can also use the ALTER TABLESPACE statement to have an existing table space inherit media attributes from its storage group.

When a table space dynamically inherits media attributes from its associated storage group and if the table space uses a new storage group, then the table space media attributes are dynamically set to the attributes of the new storage group.

This allows for easier storage management, as the media attributes do not have to be explicitly specified whenever a new storage group is used.

When creating a storage group, you can specify the following media attributes:

OVERHEAD

This attribute specifies the I/O controller overhead and the disk seek and latency time in milliseconds.

DEVICE READ RATE

This attribute specifies the device specification for the read transfer rate in megabytes per second. This value is used to determine the cost of I/O during query optimization. If this value is not the same for all storage paths, the number should be the average for all storage paths that belong to the storage group.

DATA TAG

This attribute specifies a tag on the data in a particular storage group, which WLM can use to determine the processing priority of database activities.

The default values for the storage group attributes are as follows:

Table 1. The default settings for storage group attributes

Attribute	Default setting
DATA TAG	NONE
DEVICE READ RATE	100 MB/sec
OVERHEAD	6.725 ms

When you create or alter an automatic storage table space, you can specify the storage group name along with the DATA TAG, TRANSFERRATE, and OVERHEAD attributes. However, the table space can dynamically inherit these attributes from the associated storage group by specifying the INHERIT clause.

Note: The TRANSFERRATE attribute inherits the DEVICE READ RATE attribute value from the storage group if the TRANSFERRATE INHERIT clause is specified.

Related concepts:

"Storage group and table space media attributes" in Database Administration Concepts and Configuration Reference

New mechanism to reclaim unused space in indexes

A new mechanism to reclaim unused space in indexes has been introduced to provide a more efficient way to free space for indexes that reside in DMS table spaces.

Deleting a substantial amount of data from tables on a regular basis results in unused space in tables and associated indexes. This space cannot be used by any other object in the same table space until reorganization takes place.

In Version 10.1, you can use the new online index reorganization functionality to reclaim unused index space on tables that reside in DMS table spaces. This functionality is available through the following options:

- Issuing the **REORG INDEX FOR TABLE** or **REORG INDEXES ALL FOR TABLE** command with the new RECLAIM EXTENTS clause.

- Calling the db2Reorg API and specifying the new DB2REORG_INDEX_RECLAIM_EXTENTS value for the reorgFlags parameter in the db2ReorgStruct data structure.
- Setting automatic index reorganization and specifying the reclaimExtentsSizeForIndexObjects attribute in the ReorgOptions element in the input XML file.

Related reference:

"REORG INDEXES/TABLE " in Command Reference

New insert time clustering tables

Insert time clustering (ITC) tables provide an effective way of maintaining data clustering and easier management of space utilization.

ITC tables have similar characteristics to MDC tables. For example, these table types use block based allocation and block indexes. ITC and MDC tables differ in how data is clustered. ITC tables cluster data by using a virtual column which clusters rows, that are inserted at a similar time, together. Clustering dimensions on MDC tables are specified by the creator.

ITC tables are created with the CREATE TABLE command by specifying the ORGANIZE BY INSERT TIME clause.

A convenient, online way to convert existing tables to ITC tables is the ADMIN_MOVE_TABLE procedure. Another method to convert existing tables to ITC tables is export/import or a load from table. Existing tables cannot be altered to become ITC tables.

Related concepts:

"Scenario: ExampleBANK reclaiming table and index space" in Troubleshooting and Tuning Database Performance

"The db2move command now supports parallel processing"

Related tasks:

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

The db2move command now supports parallel processing

In specific cases, the **db2move** command supports parallel processing.

If COPY mode is specified when copying a schema, the **db2move** command can now load the tables in the schema in parallel by using the new PARALLEL option.

Related reference:

"db2move - Database movement tool " in Command Reference

Partitioned tables remain accessible to queries when a data partition is added or attached

The process of adding or attaching a data partition to a partitioned table by using the ALTER TABLE statement with the ADD PARTITION or ATTACH PARTITION clause has been enhanced. The partitioned table now remains accessible to dynamic queries running under the RS, CS, or UR isolation level.

Moreover, if you perform data integrity checking before an attach operation, the newly attached data can be made available much sooner. You can optimize the data roll-in process by using the SET INTEGRITY... ALL IMMEDIATE UNCHECKED statement to skip unnecessary range and constraints violation checking. In this case, the table is brought out of SET INTEGRITY pending state, and the new data is available for applications to use immediately if there are no nonpartitioned user indexes on the target table.

Related concepts:

"Table partitioning" in Partitioning and Clustering Guide

Related reference:

"ALTER TABLE " in SQL Reference Volume 2

"SET INTEGRITY " in SQL Reference Volume 2

Chapter 4. pureXML enhancements

DB2 V10.1 supports the XML data type and enhances the pureXML feature to make your data processing even more flexible, faster, and more reliable.

In DB2 V10.1, the pureXML feature is enhanced by the following performance and ease-of-use improvements:

- New XML indexes more closely match your data (see “New types supported for indexes over XML”)
- Functional indexes can speed up searches and queries (see “Functional XML indexes can speed up queries”)
- Binary XML format enables faster data transmission (see “New binary XML format improves performance for certain Java clients” on page 19)
- Handling of casting errors is now aligned with SQL (see “XML casting and truncation errors reduced” on page 20)
- Performance improved for XMLTABLE function queries (see “Performance improved for certain XML queries” on page 22)

New types supported for indexes over XML

You can now create indexes of type DECIMAL and INTEGER over XML data. In situations where your numeric data is of either INTEGER or DECIMAL type, indexes created as DECIMAL and INTEGER values can potentially provide faster query response times.

In previous releases, DOUBLE was the only supported numeric type for XML indexes. Unbounded decimal values and 64-bit integers might lose precision when stored in DOUBLE indexes, which can lead to slower query performance for DOUBLE indexes for this kind of data. You can avoid this potential slower performance by using the new index types of INTEGER and DECIMAL, when appropriate for your data.

The new DECIMAL and INTEGER index types are fully supported in a partitioned database environment as either local or global indexes.

Related concepts:

"Data types associated with index XML pattern expressions" in pureXML Guide

Related reference:

"CREATE INDEX " in SQL Reference Volume 2

Functional XML indexes can speed up queries

Starting in DB2 V10.1, you can create functional XML indexes using the fn:upper-case and fn:exists functions. Indexes created using fn:upper-case can speed up case-insensitive searches of XML data. Indexes created using fn:exists can speed up queries that search for specific elements or for the lack of specific elements.

Also, in DB2 V10.1, for queries with predicates that contain the fn:starts-with function, the optimizer can now choose to use VARCHAR-type indexes.

Using indexes created with fn:upper-case for case insensitive searches

In previous releases, to search for all occurrences of a string value in a certain path independently of case, you had to use a query that converted the searched data to all one case (either upper or lower). This query did not use an XML index to speed up the search.

In DB2 V10.1, you can create a functional XML index of type VARCHAR or VARCHAR HASHED that converts string data to its uppercase form. To do this, specify fn:upper-case in the XMLPATTERN clause of the CREATE INDEX statement. For example:

```
CREATE INDEX clients_state_idx ON clients(contactinfo)
  GENERATE KEYS USING XMLPATTERN '/Client/address/state/fn:upper-case(.)'
  AS SQL VARCHAR(50);
```

The optimizer can choose to use this index for queries with predicates that match the XML path in the XMLPATTERN clause, and that also specify the fn:upper-case function, such as in the following query fragment:

```
XQUERY db2-fn:xmlcolumn('CLIENTS.CONTACTINFO')
  [Client/address/state/fn:upper-case(.)="NEW YORK"];
```

For large data sets, use of such an index can potentially provide significant performance gains.

When you create a case-insensitive index, if required, you can use the optional locale parameter of the fn:upper-case function. For example, the following statement creates an index on the address attribute, type (with the path /Client/address/@type) for the tr_TR locale:

```
CREATE INDEX client_address_type_idx_tr ON clients(contactinfo)
  GENERATE KEYS USING XMLPATTERN '/Client/address/@type/fn:upper-case(., "tr_TR")'
  AS SQL VARCHAR(50);
```

For the client_address_type_idx_tr index to be considered by the optimizer, a query must also specify the same locale, match the XML path in the XMLPATTERN clause, and specify the fn:upper-case function.

Using indexes created with fn:exists to search only elements or attributes that exist

In DB2 V10.1, you can create an XML index that checks for the existence of an element or attribute by including fn:exists in the XMLPATTERN clause of the CREATE INDEX statement. The element or attribute must be specified as the parameter of fn:exists, and the index must be of type VARCHAR(1).

For example, the following index stores a single character, T or F, to indicate whether it is true or false that an employee has a middle name recorded in the XML document structure:

```
CREATE INDEX empindex on company(companydocs)
  GENERATE KEY USING XMLPATTERN
  '/company/emp/name/fn:exists(middle)' AS SQL VARCHAR(1);
```

The fn:exists function checks for the existence, or lack of existence, of a specific element.

The optimizer can choose to use this index for queries that search for the parameter of `fn:exists`, in this case, the middle name, as in the following query fragment:

```
XQUERY db2-fn:xmlcolumn('COMPANY.COMPANYDOCS')
      /company/emp/name[fn:exists(middle)];
```

Using VARCHAR indexes with queries having predicates that contain `fn:starts-with`

In DB2 V10.1, for queries with predicates that contain `fn:starts-with`, the optimizer can choose to use VARCHAR-type indexes to speed up the query. No changes are necessary to existing VARCHAR indexes, or use of any special syntax in the CREATE INDEX statement for new indexes. In previous releases, queries with predicates that contained the `fn:starts-with` function did not use XML indexes for access and had to use table scans.

The `fn:starts-with` function determines whether a string begins with a specific substring.

Related concepts:

"Index XML pattern expressions" in pureXML Guide

"Examples of using case insensitive XML indexes" in pureXML Guide

"Examples of using indexes that specify `fn:exists`" in pureXML Guide

Related reference:

"CREATE INDEX " in SQL Reference Volume 2

"exists function" in XQuery Reference

"upper-case function" in XQuery Reference

New binary XML format improves performance for certain Java clients

The new binary XML format provides a faster way to transmit and receive XML data between certain Java pureXML applications and a DB2 server Version 10.1. For these Java applications, unnecessary XML parsing costs are eliminated, therefore improving performance.

Binary XML data refers to data that is in the Extensible Dynamic Binary XML DB2 Binary XML Format, also known as XDBX format.

For JDBC and SQLJ applications, you can now choose to transmit data to and from a DB2 server Version 10.1 in binary XML format. For applications that work with data in a non-textual representation, such as those using SAX or StAX objects, the binary format provides a faster way to transmit and receive XML data. In earlier releases, only textual XML data format was supported. Now, you can use whichever format best suits your data processing needs. Binary XML format is only used for data transmission. You will not see data in binary format stored in the database or anywhere else.

For JDBC and SQLJ applications that work with data in a non-textual representation, binary XML format eliminates unnecessary XML parsing and serialization costs, therefore improving performance. For example, you should see significant performance improvements if your application uses any of the following methods to retrieve and update XML data:

- `getSource(SAXSource.class)`, `getSource(StAXSource.class)`
- `setResults(SAXResults.class)`, `setResults(StAXResult.class)`

The degree of performance improvement also depends on the structure of the XML documents, the length of tags, the number of repeating tags, and the depth of data within the document.

To use the new binary XML format, you need to use Version 4.9, or later, of the IBM Data Server Driver for JDBC and SQLJ to connect to a DB2 V10.1, or later, server. For SQLJ applications, you also need to use Version 4.9, or later, of the sqlj4.zip package.

For JDBC and SQLJ applications that use Version 4.9, or later, of the IBM Data Server Driver for JDBC and SQLJ, binary XML is the default format when the application connects to a DB2 server Version 10.1 or later release. You can use the `xmlFormat` property in the `DriverManager` and `DataSource` interfaces to control whether the transmission of XML data is in textual or binary format.

You can use binary XML format with any valid SQL/XML or XQuery statements.

Related concepts:

"Binary XML format in Java applications" in pureXML Guide

XML casting and truncation errors reduced

Starting in DB2 V10.1, the DB2 database manager handles XML casting in a similar way as for SQL. Situations that used to raise errors now allow applications to continue. Also, to enhance usability, queries that compare XML data of incompatible types return FALSE instead of returning error SQL16061N.

Changes in casting to a CHAR or VARCHAR type

In DB2 V10.1, as in SQL processing, casting XML data to a CHAR or VARCHAR type that is too small causes the data to be truncated to fit the specified data type and no error is returned. If non-blank characters are truncated, warning SQL0445W is returned. In previous releases, casting XML data to a CHAR or VARCHAR type that was too small caused error SQL16061N to be returned.

For example, in earlier releases, the following query fragment that casts the 14 character string 'SQL standards ' to a CHAR(13) data type causes error SQL16061N to be returned:

```
VALUES XMLCAST(XMLQUERY('SQL standards ') AS char(13));
```

Results:

```
SQL016061N The value "SQL standards " cannot be constructed as, or cast
(using an implicit or explicit cast) to the data type char(13).
```

For the same query fragment running in DB2 V10.1, the data is truncated without an error. Because the truncated 14th character is a blank character, no warning message is issued.

```
VALUES XMLCAST(XMLQUERY('SQL standards ') AS char(13));
```

Results:

```
1
-----
SQL standards
```

1 record(s) selected.

In DB2 V10.1, if the 13 character string 'SQL standards' is cast to a VARCHAR(12) data type, no error is returned. However, because the character 's' was truncated, warning message SQL0445W is issued.

```
VALUES XMLCAST(XMLQUERY('"SQL standards"') AS varchar(12));
```

Results:

```
1
-----
SQL standard
SQL0445W Value "SQL standards" has been truncated.  SQLSTATE=01004
```

1 record(s) selected with 1 warning messages printed.

This change in behavior might also occur when you use the XMLTABLE function. The XMLTABLE function allows you to execute an XQuery expression and return values as a table instead of as a sequence of values. In the COLUMNS clause of the XMLTABLE function, you define the characteristics of each column, such as the data type. For CHAR and VARCHAR columns, if any non-blank characters are truncated, the XMLTABLE function returns warning SQL0445W.

Changes in casting to a DECIMAL type

In DB2 V10.1, as in SQL processing, if you cast XML data to a DECIMAL type that has insufficient space for digits to the right of the decimal separator, the trailing digits are truncated to fit the specified data type and no error is returned. In previous releases, error SQL16061N was returned.

As in previous releases, if the data value overflows the specified DECIMAL type (the type has insufficient space for digits to the left of the decimal separator) error SQL16061N continues to be returned.

The DECIMAL type takes two parameters, *precision* and *scale*. The first parameter, *precision*, is an integer constant with a value in the range of 1 to 31 that specifies the total number of digits. The second parameter, *scale*, is an integer constant that is greater than or equal to zero, and less than or equal to *precision*. The *scale* specifies the number of digits to the right of the decimal point.

The following example shows what happens in DB2 V10.1 when various values are cast to a DECIMAL(3,2) type:

Table 2. Examples of casting values to DECIMAL(3,2) and their results

Value	Result	Comments
1.0	1.0	No truncation occurs
3.23	3.23	No truncation occurs
0.2	0.2	No truncation occurs
9.99	9.99	No truncation occurs
1.056	1.05	All digits to the right of the second decimal position are truncated. No error or warning is issued.
3.230	3.23	All digits to the right of the second decimal position are truncated. No error or warning is issued.
0.006	0.00	All digits to the right of the second decimal position are truncated. No error or warning is issued.
9.9999	9.99	All digits to the right of the second decimal position are truncated. No error or warning is issued.

Table 2. Examples of casting values to DECIMAL(3,2) and their results (continued)

Value	Result	Comments
19.9	Error returned.	This value overflows the DECIMAL(3,2) type. Error SQL16061N is raised.
165	Error returned.	This value overflows the DECIMAL(3,2) type. Error SQL16061N is raised.
99.678	Error returned.	This value overflows the DECIMAL(3,2) type. Error SQL16061N is raised.

This change in casting behavior can also occur when you use the XMLTABLE function. The XMLTABLE function converts XML values to the data types of the target columns that you are creating.

Changes in comparisons

In DB2 V10.1, if your query compares XML data of incompatible types, the comparison returns FALSE. Previously, error SQL16061N was returned.

For example, in earlier releases, because the following query compares the character string 'N/A' with the number, 3.4, error SQL16061N is returned:

```
xquery let $doc := <a><b>N/A</b></a> return $doc[b < 3.4];
```

Results:

```
SQL016061N The value "N/A" cannot be constructed as, or cast (using an implicit or explicit cast) to the data type double.
```

For the same query running in DB2 V10.1, no error is raised. The comparison results in FALSE, therefore no rows are returned for this query:

```
xquery let $doc := <a><b>N/A</b></a> return $doc[b < 3.4];
```

Results:

```
1
-
0 record(s) selected.
```

Related reference:

"Casting between data types" in SQL Reference Volume 1

"General comparisons" in XQuery Reference

Performance improved for certain XML queries

In DB2 V10.1, the DB2 server has been optimized to improve performance for certain commonly used queries, such as those that use the XMLTABLE function.

Examples of queries that might show faster response are as follows:

- **Queries that use the XMLTABLE function.** For example:

```
SELECT T.* FROM TEST,
XMLTABLE('$doc/a/b' passing TEST.XMLCOL as "doc" columns
         c varchar(10) path 'c1/c2/c'
         d varchar(10) path 'd1/d2/d'
         e varchar(10) path 'e1/e2/e') AS T;
```

- **Non-linear XQuery queries** (with multiple paths, or branches). For example:

```
xquery for $a in db2-fn:xmlcolumn('XTAB.DOC')/a
       for $b in $a/b
       for $c in $a/c
       return <res>{$b,$c}</res>
```

- **Queries with early-out join predicates.** An early-out join is a join in which at most one row from the inner table must match a row in the outer table. For example, the following XMLTABLE query has an early-out join on a row generator:

```
SELECT stat, gen FROM custacc,
       XMLTABLE('$CADOC/Customer [DateOfBirth >= xs:date("1910-01-01")
              and BankingInfo/PremiumCustomer = "No"] '
       COLUMNS
         GEN  VARCHAR(20) PATH 'Gender',
         Nationality VARCHAR(20) PATH 'Customer/Nationality',
         STAT VARCHAR(20) PATH 'BankingInfo/CustomerStatus');
```

The next example has an early out join on the for clause:

```
xquery for $i in db2-fn:xmlcolumn('T.XMLCOL')/PRODUCT,
       $j in $i[NAME='5Z761']//PRICE
return $j
```

- **Queries with a parent axis.** For example, the following XMLTABLE query has a parent axis that is not on the output step:

```
SELECT T.* FROM TEST,
       XMLTABLE ( '$doc/a/b' passing TEST.XMLCOL as "doc" columns
         c varchar(10) path 'c1/c2/c'
         d varchar(10) path '../d1/d2/d'
         e varchar(10) path '../../e1/e2/e') as T
```

The next example also has a parent axis that is not on the output step:

```
xquery let $doc := db2-fn:xmlcolumn('T.XMLCOL') return ($doc/root//a/.. )/b
```

Chapter 5. Monitoring enhancements

Version 10.1 includes enhancements that make monitoring DB2 database environments more comprehensive with higher granularity of control.

The following monitoring enhancements provide new monitoring information:

- Event monitor that tracks configuration changes (see “New event monitor tracks configuration and registry changes and DDL and utility executions”)
- Usage list object keeps track of statements that affect a table or index (see “New usage list object identifies statements that affect tables or indexes” on page 26)
- Create thresholds for specific statements with the new STATEMENT threshold domain (see “New STATEMENT domain for thresholds lets you define thresholds for statements that contain specific text” on page 26)
- New and changed table functions for accessing monitoring information (see “New and changed functions for accessing monitoring information” on page 27)
- List of executable IDs now included in information captured by unit of work event monitor (see “Executable ID list is included in unit of work event monitor information” on page 31)
- Write-to-table support for all event monitors (see “All event monitors now support the WRITE TO TABLE target” on page 31)
- Existing write-to-table event monitors can be altered to capture additional logical data groups (see “Existing event monitors that write to tables can be altered to capture additional logical data groups” on page 32)
- Upgrade of event monitor output tables that were created in previous releases (see “Event monitor tables can now be upgraded” on page 33)
- Pruning data from unformatted event tables (see “Pruning data from unformatted event tables is now supported” on page 33)
- New monitor elements that provide greater insight into DB2 servers (see “New monitor elements provide added insight into the operation of your DB2 server” on page 34)

New event monitor tracks configuration and registry changes and DDL and utility executions

The change history event monitor captures changes to database and database manager configuration and registry settings, execution of DDL statements, and execution of utilities. You can use this data to determine if the appearance of any problems with your database coincides with any of these events.

Some examples of changes to the system that might affect the performance or behavior of work on the system include:

- Creating or dropping an index unexpectedly
- The failure of scheduled maintenance to run
- Changes to a database configuration parameter or DB2 registry setting

Unintentional or unanticipated changes affecting database operations might be made by a user; for example, a DBA might execute DDL to drop an index. Or, changes might occur automatically, without any user interaction; for example, the self-tuning memory manager (STMM) might change a configuration parameter, or

automatic table reorganization might reorganize a table. Either way, you can use the change history event monitor to track several different types of changes, including:

- Database and database manager configuration parameter changes
- Registry variable changes
- Execution of DDL statements
- Execution of utilities (for example, RUNSTATS, LOAD, REORG)

The change history event monitor can record certain of types changes made even when the database is offline.

Related concepts:

"Change history event monitoring" in Database Monitoring Guide and Reference

New usage list object identifies statements that affect tables or indexes

Use the new *usage list* database object to record the DML statement sections that reference a particular table or index and capture statistics about how those sections affect each object as they execute.

Each entry in the usage list includes information about the number of times that a section executed over a particular time frame. The entries also contain aggregate statistics that indicate how the section affected the table or index across all executions.

The usage list also includes statistics about factors such as lock and buffer pool usage for each statement section. If you determine that a statement negatively affected a table or index, use these statistics to determine where further monitoring might be required or how you can tune the statement.

Related concepts:

"Usage lists" in Database Administration Concepts and Configuration Reference

Related tasks:

"Identifying the statements that affect a table" in Database Monitoring Guide and Reference

Related reference:

"CREATE USAGE LIST " in SQL Reference Volume 2

New STATEMENT domain for thresholds lets you define thresholds for statements that contain specific text

In DB2 Version 10.1, a new threshold domain called STATEMENT has been added to the CREATE THRESHOLD statement syntax. This domain makes it possible to define thresholds for execution of specific statements.

For example, you can define a CPUTIME threshold for a SQL statement like "SELECT * FROM TABLE1, TABLE2" so that a threshold violation occurs when this statement is executed and the CPU time threshold for the statement is exceeded. You can identify the statement for these thresholds by specifying either the statement text, as in this example, or the executable ID for the statement. Similar to thresholds on other domains, you can configure STATEMENT thresholds to write information about activities that violate the threshold to the activity event monitor.

This new capability provides greater granularity and specificity in capturing information than in earlier releases. In previous releases, identifying problems with activities for a specific statement required that you capture information for many activities, and then sift through the event monitor data to look for anomalies. Now, when you identify a statement that runs for longer than expected, you can quickly collect and examine activity information related to just that statement. For example, you can view the data represented by parameter markers that indicate a product identifier in the statement. Or, you might find it useful to examine time-spent monitor elements related to the execution of the statement, such as the total execution time (TOTAL_EXEC_TIME).

Related tasks:

"Example: Capturing activity information related to the execution of a statement" in Database Monitoring Guide and Reference

Related reference:

"CREATE THRESHOLD " in SQL Reference Volume 2

New and changed functions for accessing monitoring information

Several table functions and two scalar functions have been added and several table functions have been extended so that you can retrieve additional monitoring information using SQL.

The following table describes the new table functions in Version 10.1 that return monitoring information:

Table 3. New table functions that return monitoring information

Name	Details
ADMIN_GET_STORAGE_PATHS	Returns a list of automatic storage paths for each database storage group and file system information for each storage path.
MON_GET_AUTO_MAINT_QUEUE	Returns information about all automatic maintenance jobs that are currently queued for execution by the autonomic computing daemon (db2acd) except for realtime statistics jobs.
MON_GET_AUTO_RUNSTATS_QUEUE	Returns information about all objects that are currently queued for evaluation by automatic statistics collection in the currently connected database.
MON_GET_CF	Returns status information about one or more cluster caching facilities on the system.
MON_GET_CF_CMD	Reports the amount of time, in microseconds, that the cluster caching facilities spends processing a request.
MON_GET_CF_WAIT_TIME	Reports the amount of time, in microseconds, spent waiting for the cluster caching facilities to process a request and the amount of time, in microseconds, spent for related communications with the cluster caching facilities.
MON_GET_EXTENDED_LATCH_WAIT	Returns information about latches that have been involved in extended waits.

Table 3. New table functions that return monitoring information (continued)

Name	Details
MON_GET_GROUP_BUFFERPOOL	Returns information about the group buffer pool.
MON_GET_HADR	Returns high availability disaster recovery information.
MON_GET_INDEX_USAGE_LIST	Returns information from the usage list that is defined for an index.
MON_GET_MEMORY_SET	Returns metrics from the allocated memory sets, both at the instance level, and for all active databases within the instance.
MON_GET_MEMORY_POOL	Returns metrics from the memory pools contained within a memory set.
MON_GET_PAGE_ACCESS_INFO	Returns information about buffer pool pages that are being waited on for a specified table.
MON_GET_REBALANCE_STATUS	Returns the status of a rebalance operation on a table space.
MON_GET_RTS_RQST	Returns information about all realtime statistics requests that are pending in the system and the set of requests that are currently being processed by the realtime statistics daemon.
MON_GET_SERVERLIST	Returns metrics on the server list for the currently connected database as cached on one or more members.
MON_GET_TABLE_USAGE_LIST	Returns information from the usage list that is defined for a table.
MON_GET_TRANSACTION_LOG	Returns information about the transaction logging subsystem for the currently connected database.
MON_GET_USAGE_LIST_STATUS	Returns information about a usage list, such as how big it is, when it was last changed, and how much memory is allocated for the list.
MON_SAMPLE_SERVICE_CLASS_METRICS	Reads system metrics from one or more service classes across one or more databases at two points in time and computes various statistics from these metrics.
MON_SAMPLE_WORKLOAD_METRICS	Reads system metrics from one or more workloads across one or more databases at two points in time and computes various statistics from these metrics.

The following table describes the new scalar functions in Version 10.1 that return monitoring information:

Table 4. New scalar functions that return monitoring information

Name	Details
MON_GET_APPLICATION_HANDLE	Returns the application handle of the invoking application.

Table 4. New scalar functions that return monitoring information (continued)

Name	Details
MON_GET_APPLICATION_ID	Returns the application ID of the invoking application.

The following table describes the table functions that have been changed in Version 10.1 to return additional monitoring information:

Table 5. Table functions that return additional monitoring information

Name	Details
MON_BP_UTILIZATION	Returns additional columns such as AVG_ASYNC_READ_TIME, AVG_ASYNC_WRITE_TIME, AVG_SYNC_READ_TIME, AVG_SYNC_WRITE_TIME,, and GBP_XDA_HIT_RATIO_PERCENT.
MON_GET_ACTIVITY_DETAILS	Returns additional columns such as columns that report information about data tags in service class thresholds.
MON_GET_BUFFERPOOL	Returns additional columns such as columns that report metrics about asynchronous reading from the buffer pool and writing to the buffer pool and I/O server efficiency.
MON_GET_CONNECTION and MON_GET_CONNECTION_DETAILS	Returns additional columns that provide information such as this: <ul style="list-style-type: none"> • Metrics about the efficiency of I/O servers • The number of completed non-nested activities that are submitted from external applications • The amount of processing time for authentication, statistics generation, statement execution, and extended latch waits
MON_GET_CONTAINER	Returns an additional column describing the storage path ID for a container.
MON_GET_INDEX	Returns additional columns such as columns that report metrics on reading from buffer pools and writing to buffer pools.
MON_GET_PKG_CACHE_STMT and MON_GET_PKG_CACHE_STMT_DETAILS	Return the routine identifier that is associated with the target of a CALL statement. For other statements, the value is 0. The function also returns additional columns that report metrics about I/O server efficiency, processing time for authentication, statistics generation, statement execution, high water mark input values, and extended latch waits.

Table 5. Table functions that return additional monitoring information (continued)

Name	Details
MON_GET_SERVICE_SUBCLASS and MON_GET_SERVICE_SUBCLASS_DETAILS	<p>Returns additional columns that provide information such as this:</p> <ul style="list-style-type: none"> • Metrics about the efficiency of I/O servers • The number of completed non-nested activities that are submitted from external applications • The amount of processing time for authentication, statistics generation, statement execution, and extended latch waits
MON_GET_TABLE	<p>Returns additional columns such as columns that report metrics on reading from buffer pools and writing to buffer pools.</p>
MON_GET_TABLESPACE	<p>Returns additional columns that report information such as this:</p> <ul style="list-style-type: none"> • The number of the last consecutive page at the start of the object table • Metrics for asynchronous reading from buffer pools and writing to buffer pools • Storage group information • Data tag information
MON_GET_UNIT_OF_WORK and MON_GET_UNIT_OF_WORK_DETAILS	<p>Returns additional columns that provide information such as this:</p> <ul style="list-style-type: none"> • Metrics about the efficiency of I/O servers • The number of completed non-nested activities that are submitted from external applications • The amount of processing time for authentication, statistics generation, statement execution, and extended latch waits
MON_GET_WORKLOAD and MON_GET_WORKLOAD_DETAILS	<p>Returns additional columns that provide information such as this:</p> <ul style="list-style-type: none"> • Metrics about the efficiency of I/O servers • The number of completed non-nested activities that are submitted from external applications • The amount of processing time for authentication, statistics generation, statement execution, and extended latch waits
WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES	<p>Returns the APPL_ID column in addition to the <i>application_handle</i> value.</p>
WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES	<p>Returns additional columns such as the ENTRY_TIME column, which can help in canceling activities that are queued for more than a certain amount of time.</p>

Related concepts:

"Monitor routines" in Administrative Routines and Views

"New usage list object identifies statements that affect tables or indexes" on page 26

Executable ID list is included in unit of work event monitor information

You can now collect the executable ID list and associated statement-level metrics for a unit of work. Inclusion of an executable ID list can help facilitate SQL statement troubleshooting.

You can enable the collection of this information by using one of the following two mechanisms:

- Enable collection at the database level by setting the `mon_uow_data` database configuration parameter to `BASE` and the `mon_uow_execlist` database configuration parameter to `ON`, as shown in the following example:

```
UPDATE DB CFG FOR SAMPLE USING mon_uow_data BASE
UPDATE DB CFG FOR SAMPLE USING mon_uow_execlist ON
```
- Enable collection for a specific workload by specifying the `COLLECT UNIT OF WORK DATA` clause for the `CREATE WORKLOAD` or `ALTER WORKLOAD` statement. The syntax of the clause has changed. For details, see "ALTER WORKLOAD and CREATE WORKLOAD statements have changed".

In a partitioned database environment, the executable ID list is collected for each coordinator member or data member. In a DB2 pureScale environment, the executable ID list is collected from the coordinator member.

Related concepts:

"Unit of work event monitoring" in Database Monitoring Guide and Reference

"Executable list information" in Database Monitoring Guide and Reference

Related reference:

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

"`mon_uow_execlist` - Monitoring unit of work events with executable list configuration parameter" in Database Administration Concepts and Configuration Reference

All event monitors now support the WRITE TO TABLE target

In earlier releases, the event data of some event monitors was written to unformatted event (UE) tables, which required post-processing to be viewed. Now, all event monitors can write event data directly to relational tables.

Three new event monitors were introduced in DB2 Version 9.7:

- Locking event monitor
- Package cache event monitor
- Unit of work event monitor.

In Version 9.7, each of these event monitors wrote their output to unformatted event (UE) tables. However, because most of the event data they capture is stored in binary format, you were required to post-process the UE table to work with the data. Starting in Version 10.1, you can choose to have these event monitors write directly to relational tables, or to UE tables, whichever you prefer.

Related concepts:

"Lock and deadlock event monitoring" in Database Monitoring Guide and Reference

"Unit of work event monitoring" in Database Monitoring Guide and Reference

"Package cache statement eviction event monitoring" in Database Monitoring Guide and Reference

"Output options for event monitors" in Database Monitoring Guide and Reference

Related tasks:

"Creating event monitors that write to tables" in Database Monitoring Guide and Reference

Existing event monitors that write to tables can be altered to capture additional logical data groups

When you create an event monitor that writes to tables, you can specify that data from one or more logical data groups be excluded from the event monitor output. Starting in DB2 Version 10.1, you can use the new ALTER EVENT MONITOR statement to add logical data groups that were previously excluded from an event monitor.

In earlier releases, to add data groups that were previously excluded, you had to drop and recreate the event monitor.

For example, if you create a locking event monitor that writes to a table, you can specify that only elements from the lock_participants logical data group be captured. In this case, the event monitor creates only the LOCK_PARTICIPANTS_*evmon-name* table, where *evmon-name* is the name given to the event monitor.

If you later decide that you want to add the lock_participant_activities logical data group to this event monitor, you can use the ALTER EVENT MONITOR statement:

```
ALTER EVENT MONITOR evmon-name ADD LOGICAL GROUP lock_participant_activities
```

This statement adds a table called LOCK_PARTICIPANT_ACTIVITIES_*evmon-name* for the newly-added logical data group. It also modifies the event monitor so that it collects data from the lock_participant_activities logical data group in addition to whatever data was previously collected.

Restriction: The ALTER EVENT MONITOR statement can be used only to add logical data groups to an event monitor. You cannot remove or drop a logical data group once it has been added, nor can you change the name, target table space or the value for PCTDEACTIVATE associated with the table used to capture the data in monitor elements that belong to a data group.

Related tasks:

"Altering an event monitor" in Database Monitoring Guide and Reference

Related reference:

"ALTER EVENT MONITOR " in SQL Reference Volume 2

Event monitor tables can now be upgraded

You can now upgrade existing event monitor tables from previous releases. In previous releases, if you wanted to retain the data in your existing event monitor tables when you upgraded your DB2 product, you had to manually alter those tables to match the definitions of event monitor tables in the new release.

You can now upgrade existing target tables for event monitors that write to tables and to unformatted event (UE) tables by using the new `EVMON_UPGRADE_TABLES` procedure. This procedure performs the following tasks to upgrade your existing target tables and to add any new target tables required to store the event monitor output:

- For event monitors that write to tables, it modifies the target tables by adding new columns, removing old columns, and modifying existing columns so that the target table can accurately collect all elements. It also creates new target tables that were introduced after the event monitor was created.
- For event monitors that write to UE tables, it modifies the target table by adding new columns and modifying existing columns so that the UE table can be properly post-processed by the `EVMON_FORMAT_UE_TO_TABLES` or `EVMON_FORMAT_UE_TO_XML` routines.

You can also upgrade existing tables produced by the `EVMON_FORMAT_UE_TO_TABLES` procedure. A new `UPGRADE_TABLES` option has been added to this procedure. If you specify this option, `EVMON_FORMAT_UE_TO_TABLES` performs the following tasks to upgrade the tables required to store the event monitor output:

- It modifies any existing tables produced by `EVMON_FORMAT_UE_TO_TABLES` by adding new columns and modifying existing columns so that the UE table can be properly post-processed.
- It creates any new tables that were introduced after the event monitor was created.

Related concepts:

"Event monitor data retention from release to release" in Database Monitoring Guide and Reference

Related reference:

"`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

"`EVMON_UPGRADE_TABLES` procedure - Upgrade event monitor target tables" in Administrative Routines and Views

Pruning data from unformatted event tables is now supported

The `PRUNE_UE_TABLES` option has been added to the `EVMON_FORMAT_UE_TO_TABLES` procedure to delete data from an unformatted event (UE) table after this data has been successfully exported to relational tables.

Pruning data from UE tables is useful if you no longer need to keep data in the UE table after you have exported it with the `EVMON_FORMAT_UE_TO_TABLES` procedure. For example, if you collect data on a daily basis and then export it to regular tables for preparing reports, you might not want to keep the data in the UE table.

The `PRUNE_UE_TABLES` option only deletes data from the UE table after this data has been successfully inserted into the relational table by the `EVMON_FORMAT_UE_TO_TABLES` procedure.

Related tasks:

"Pruning data from UE tables" in Database Monitoring Guide and Reference

Related reference:

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

New monitor elements provide added insight into the operation of your DB2 server

Many new monitor elements have been added in Version 10.1.

These monitor elements report on various things, including the following ones:

- The operation of I/O servers (prefetchers) (see Table 6)
- The status of non-nested activities that are submitted by applications (see Table 7 on page 37)
- Information about `DATATAGINSC` thresholds (see Table 8 on page 38)
- Information about storage groups (see Table 9 on page 38)
- Workload monitoring information (see Table 10 on page 39)
- Time spent during connection and authentication activities (see Table 11 on page 40)
- Details that are related to the longest-running SQL statement in the package cache (see Table 12 on page 40)
- Additional measures of time spent in the system (see Table 13 on page 40)
- Buffer pool and group buffer pool activity in DB2 pureScale environments (see Table 14 on page 41)
- Information about usage lists (see Table 15 on page 43)
- Information about memory pool and memory set usage (see Table 16 on page 43)

In addition, several miscellaneous monitor elements have been added. See Table 17 on page 44.

The following table lists the new monitor elements that report on the efficiency of I/O servers (prefetchers).

Table 6. New monitor elements for data prefetching

Name	Description
pool_failed_async_data_reqs	The number of times that an attempt to queue a data prefetch request failed. One possible reason is the prefetch queue was full and a request could not be obtained from the free list.

Table 6. New monitor elements for data prefetching (continued)

Name	Description
pool_failed_async_index_reqs	The number of times that an attempt to queue an index prefetch request failed. One possible reason is the prefetch queue was full and a request could not be obtained from the free list.
pool_failed_async_other_reqs	The number of times that an attempt to queue a non-prefetch request failed.
pool_failed_async_temp_data_reqs	The number of times that an attempt to queue a data prefetch request for temporary table spaces failed.
pool_failed_async_temp_index_reqs	The number of times that an attempt to queue an index prefetch request for temporary table spaces failed.
pool_failed_async_temp_xda_reqs	The number of times that an attempt to queue an XML storage object request for temporary table spaces failed.
pool_failed_async_xda_reqs	The number of times that an attempt to queue an XML storage object request failed.
pool_queued_async_data_pages	The number of data pages that were successfully requested for prefetching.
pool_queued_async_data_reqs	The number of data prefetch requests that were successfully added to the prefetch queue.
pool_queued_async_index_pages	The number of index pages that were successfully requested for prefetching.
pool_queued_async_index_reqs	The number of index prefetch requests that were successfully added to the prefetch queue.
pool_queued_async_other_reqs	The number of requests for non-prefetch work that were successfully added to the prefetch queue.
pool_queued_async_temp_data_pages	The number of data pages for temporary table spaces that were successfully requested for prefetching.
pool_queued_async_temp_data_reqs	The number of data prefetch requests for temporary table spaces that were successfully added to the prefetch queue.
pool_queued_async_temp_index_pages	The number of index pages for temporary table spaces that were successfully requested for prefetching.
pool_queued_async_temp_index_reqs	The number of index prefetch requests for temporary table spaces that were successfully added to the prefetch queue.
pool_queued_async_temp_xda_pages	The number of XML storage object data pages for temporary table spaces that were successfully requested for prefetching.

Table 6. New monitor elements for data prefetching (continued)

Name	Description
pool_queued_async_temp_xda_reqs	The number of XML storage object data prefetch requests for temporary table spaces that were successfully added to the prefetch queue.
pool_queued_async_xda_pages	The number of XML storage object data pages that were successfully requested for prefetching.
pool_queued_async_xda_reqs	The number of XML storage object data prefetch requests that were successfully added to the prefetch queue.
pool_sync_data_gbp_reads	In a DB2 pureScale environment, the number of times that a data page was expected to be in the local buffer pool but was instead retrieved from the group buffer pool. This value is 0 in other environments.
pool_sync_data_reads	The number of times that a data page was expected to be in the buffer pool but was instead read from disk.
pool_sync_index_gbp_reads	In a DB2 pureScale environment, the number of times that an index page was expected to be in the local buffer pool but was instead retrieved from the group buffer pool. This value is 0 in other environments.
pool_sync_index_reads	The number of times that an index page was expected to be in the buffer pool but was instead read from disk.
pool_sync_xda_gbp_reads	In a DB2 pureScale environment, the number of times that an XML page was expected to be in the local buffer pool but was instead retrieved from the group buffer pool. This value is 0 in other environments.
pool_sync_xda_reads	The number of times that an XML page was expected to be in the buffer pool but was instead read from disk.
prefetch_waits	The number of times that an agent waited for an I/O server to finish loading pages into the buffer pool.
skipped_prefetch_data_p_reads	The number of data pages that an I/O server skipped because the pages were already loaded into the buffer pool.
skipped_prefetch_index_p_reads	The number of index pages that an I/O server skipped because the pages were already loaded into the buffer pool.
skipped_prefetch_temp_data_p_reads	The number of data pages for temporary table spaces that an I/O server skipped because the pages were already loaded into the buffer pool.

Table 6. New monitor elements for data prefetching (continued)

Name	Description
skipped_prefetch_temp_index_p_reads	The number of index pages for temporary table spaces that an I/O server skipped because the pages were already loaded into the buffer pool.
skipped_prefetch_temp_xda_p_reads	The number of XML storage object data pages for temporary table spaces that an I/O server skipped because the pages were already loaded into the buffer pool.
skipped_prefetch_uow_data_p_reads	The number of data pages that an I/O server skipped because the pages were already loaded into the buffer pool by a synchronous transaction.
skipped_prefetch_uow_index_p_reads	The number of index pages that an I/O server skipped because the pages were already loaded into the buffer pool by a synchronous transaction.
skipped_prefetch_uow_temp_data_p_reads	The number of data pages for temporary table spaces that an I/O server skipped because the pages were already loaded into the buffer pool by a synchronous transaction.
skipped_prefetch_uow_temp_index_p_reads	The number of index pages for temporary table spaces that an I/O server skipped because the pages were already loaded into the buffer pool by a synchronous transaction.
skipped_prefetch_uow_temp_xda_p_reads	The number of XML storage object data pages for temporary table spaces that an I/O server skipped because the pages were already loaded into the buffer pool by a synchronous transaction.
skipped_prefetch_uow_xda_p_reads	The number of XML storage object data pages that an I/O server skipped because the pages were already loaded into the buffer pool by a synchronous transaction.
skipped_prefetch_xda_p_reads	The number of XML storage object data pages that an I/O server skipped because the pages were already loaded into the buffer pool.

The following table lists the new monitor elements that count the number of non-nested activities (completed successfully, completed with errors, or rejected) that were performed from external applications.

Table 7. New monitor elements to count the number of non-nested activities

Name	Description
app_act_aborted_total	The total number of external, non-nested coordinator activities that were completed with errors.
app_act_completed_total	The total number of external, non-nested coordinator activities that were completed successfully.

Table 7. New monitor elements to count the number of non-nested activities (continued)

Name	Description
app_act_rejected_total	The total number of external, non-nested coordinator activities at any nesting level that were rejected instead of being allowed to execute.

The following table lists the new monitor elements that indicate the unique identifier for a threshold, list the data tags that were applied to a threshold, and indicator whether a threshold was violated.

Table 8. New monitor elements for thresholds

Name	Description
datataginsc_threshold_id	The ID of the DATATAGINSC IN threshold that was applied to an activity.
datataginsc_threshold_value	The list of data tags, separated by commas, in the DATATAGINSC IN threshold that was applied to an activity.
datataginsc_threshold_violated	The value that indicates whether an activity violated the DATATAGINSC IN threshold.
datatagnotinsc_threshold_id	The ID of the DATATAGINSC NOT IN threshold that was applied to an activity.
datatagnotinsc_threshold_value	The list of data tags, separated by commas, in the DATATAGINSC NOT IN threshold that was applied to an activity.
datatagnotinsc_threshold_violated	The value that indicates whether an activity violated the DATATAGINSC NOT IN threshold.

The following table lists the new monitor elements that support the monitoring of table spaces and storage groups.

Table 9. New monitor elements for table spaces and storage groups

Name	Description
db_storage_path_id	The unique identifier for each occurrence of a storage path in a storage group.
query_data_tag_list	A comma-separated list of data tag values that were referenced in a statement.
storage_group_id	An integer that uniquely represents a storage group that is used by the current database.
storage_group_name	The name of a storage group.
tablespace_rebalancer_source_storage_group_id	The source storage group identifier if the rebalancer is moving a table space from one storage group to another.
tablespace_rebalancer_source_storage_group_name	The source storage group name if the rebalancer is moving a table space from one storage group to another.
tablespace_rebalancer_target_storage_group_id	The target storage group identifier if the rebalancer is moving a table space from one storage group to another.

Table 9. New monitor elements for table spaces and storage groups (continued)

Name	Description
tablespace_rebalancer_target_storage_group_name	The target storage group name if the rebalancer is moving a table space from one storage group to another.
tbsp_datatag	The data tag value that was specified explicitly for the table space or inherited from the table space storage group.
tbsp_last_consec_page	Object relative page number of the last contiguous meta-data page for the table space.

The following table lists the new monitor elements that enhance monitoring of workloads.

Table 10. New monitor elements for workload monitoring

Name	Description
act_throughput	The number of coordinator activities per second that were completed at any nesting level.
cpu_limit	The WLM dispatcher CPU limit that is configured for a service class.
cpu_share_type	The type of WLM dispatcher CPU shares that are configured for a service class.
cpu_shares	The number of WLM dispatcher CPU shares that are configured for a service class.
cpu_utilization	The total CPU time that was consumed by a service class or workload on a particular logical partition divided by the amount of CPU time that was available on the host or the LPAR in a particular period of time.
cpu_velocity	The amount of contention for CPU resources, measured on a scale of 0 - 1, with lower numbers meaning greater contention.
estimated_cpu_entitlement	The percentage of total CPU consumption on a host or an LPAR that a service subclass is configured to consume based on its CPU shares.
total_disp_run_queue_time	The total time, in microseconds, that was spent waiting to access the CPU for requests that were run in a service class.
uow_completed_total	The total number of units of work that were completed, either by being committed or rolled back.
uow_lifetime_avg	The average lifetime of a unit of work, in milliseconds.
uow_throughput	The number of completed units of work per second.

The following table lists the new monitor elements that report connection request and authentication processing times.

Table 11. New monitor elements for connection request and authentication processing times

Name	Description
total_connect_authentication_proc_time	The amount of processing (non-wait) time that was spent performing connection or switching user authentication, in milliseconds.
total_connect_authentication_time	The amount of time that was spent performing connection or switching user authentication, in milliseconds.
total_connect_authentications	The number of connection or switching user authentications that were performed.
total_connect_request_proc_time	The amount of processing (non-wait) time that was spent processing a connection or a switch user request, in milliseconds.
total_connect_request_time	The amount of time that was spent performing a connection or a switch user request, in milliseconds.
total_connect_requests	The total number of connection or switch user requests.

The following table lists the new monitor elements that report the following items:

- The maximum statement execution time
- The input variables that were associated with that execution of the statement
- The time that the statement started executing
- The routine that was associated with that statement

Table 12. New monitor elements for package cache

Name	Description
max_coord_stmt_exec_time_args	An XML document, having parent a element with the name db2_max_coord_stmt_exec_time_args, consisting of one or more elements with the name db2_max_coord_stmt_exec_time_arg and type db2_max_coord_stmt_exec_time_arg_type.
max_coord_stmt_exec_time	The maximum coordinator execution time of a single execution of a statement, in milliseconds.
max_coord_stmt_exec_timestamp	The time that the statement that produced the max_coord_stmt_exec_time monitor element value began execution.
routine_id	The unique routine identifier that was associated with the target of a CALL statement. This monitor element returns 0 if the activity is not part of a routine.

The following table lists the new time-spent monitor elements that provide information about asynchronous reads and writes, wait times, statistics fabrications, and synchronous real-time statistics components.

Table 13. New time-spent monitor elements

Name	Description
async_read_time	The total amount of time that asynchronous engine dispatchable units (EDUs) spent reading from the buffer pool or table space.

Table 13. New time-spent monitor elements (continued)

Name	Description
async_write_time	The total amount of time that asynchronous EDUs spent writing to the buffer pool or table space.
evmon_wait_time	The amount of time that an agent waited for an event monitor record to become available. See also evmon_waits_total.
total_extended_latch_wait_time	The amount of time, in milliseconds, that was spent in extended latch waits.
total_extended_latch_waits	The number of extended latch waits.
total_stats_fabrication_proc_time	The total non-wait time that real-time statistics gathering spent on statistics fabrications, in milliseconds.
total_stats_fabrication_time	The total time that real-time statistics gathering spent on statistics fabrications, in milliseconds. See also total_stats_fabrications.
total_sync_runstats_proc_time	The non-wait time that was spent on synchronous RUNSTATS command activities that real-time statistics gathering triggered, in milliseconds.
total_sync_runstats_time	The total time that was spent on synchronous RUNSTATS command activities that real-time statistics gathering triggered, in milliseconds. See also total_sync_runstats.

The following table lists the new monitor elements that provide information about buffer pool and group buffer pool (GBP) utilization.

Table 14. New monitor elements for buffer pools and group buffer pools

Name	Description
object_data_gbp_invalid_pages	The number of times that a data page was requested for a table from the GBP because the page was invalid in the local buffer pool (LBP).
object_data_gbp_invalid_pages	The number of times that a data page was requested for a table from the GBP because the page was invalid in the LBP.
object_data_gbp_l_reads	The number of times that a GBP dependent data page was requested for a table from the GBP because the page was either invalid or not present in the LBP.
object_data_gbp_p_reads	The number of times that a GBP dependent data page was read for a table into the LBP from disk because the data page was not found in the GBP.
object_data_l_reads	The number of data pages that were requested from the buffer pool (logical) for a table.
object_data_lbp_pages_found	The number of times that a data page for a table was present in the LBP.
object_data_p_reads	The number of data pages that were read in physically for a table.
object_index_gbp_invalid_pages	The number of times that an index page was requested for an index from the GBP because the page was invalid in the LBP.

Table 14. New monitor elements for buffer pools and group buffer pools (continued)

Name	Description
object_index_gbp_l_reads	The number of times that a GBP dependent index page was requested for an index from the GBP because the page was either invalid or not present in the LBP.
object_index_gbp_p_reads	The number of times that a GBP dependent index page was read for an index into the LBP from disk because the page was not found in the GBP.
object_index_l_reads	The number of index pages that were requested from the buffer pool (logical) for an index.
object_index_lbp_pages_found	The number of times that an index page for an index was present in the LBP.
object_index_p_reads	The number of index pages that were read in physically for an index.
object_name	The object name for a table or an index. The objtype monitor element indicates whether the object is a table or an index.
object_schema	The schema name for a table or an index. The objtype monitor element indicates whether the object is a table or an index.
object_xda_gbp_invalid_pages	The number of times that a data page for an XML storage object was requested for a table from the GBP because the page was invalid in the LBP.
object_xda_gbp_l_reads	The number of times that a GBP dependent data page for an XML storage object was requested for a table from the GBP because the page was either invalid or not present in the LBP.
object_xda_gbp_p_reads	The number of times that a GBP dependent data page for an XML storage object was read for a table into the LBP from disk because the page was not found in the GBP.
object_xda_l_reads	The number of data pages for XML storage objects that were requested from the buffer pool (logical) for a table.
object_xda_lbp_pages_found	The number of times that an XML storage object data page for a table was present in the LBP.
object_xda_p_reads	The number of data pages for XML storage objects that were read in physically for a table.
pool_async_data_gbp_indep_pages_found_in_lbp	The number of GBP independent data pages found in a local buffer pool by asynchronous EDUs.
pool_async_index_gbp_indep_pages_found_in_lbp	The number of GBP independent index pages found in a local buffer pool by asynchronous EDUs.
pool_async_xda_gbp_indep_pages_found_in_lbp	The number of GBP independent XML storage object (XDA) pages found in a local buffer pool by asynchronous EDUs.
object_data_gbp_indep_pages_found_in_lbp	The number of GBP independent data pages found in a local buffer pool (LBP) by an agent.
object_index_gbp_indep_pages_found_in_lbp	The number of GBP independent index pages found in Local Buffer Pool (LBP) by agent.

Table 14. New monitor elements for buffer pools and group buffer pools (continued)

Name	Description
object_xda_gbp_indep_pages_found_in_lbp	The number of GBP independent XML storage object (XDA) data pages found in a local buffer pool (LBP) by an agent.
pool_data_gbp_indep_pages_found_in_lbp	The number of GBP independent data pages found in a local buffer pool (LBP) by an agent.
pool_index_gbp_indep_pages_found_in_lbp	The number of GBP independent index pages found in a local buffer pool (LBP) by an agent.
pool_xda_gbp_indep_pages_found_in_lbp	The number of GBP independent XML storage object (XDA) data pages found in a local buffer pool (LBP) by an agent.

The following table lists the new monitor elements that provide information about usage lists.

Table 15. New monitor elements for usage lists

Name	Description
usage_list_last_state_change	The time stamp when the value of the usage_list_state monitor element last changed.
usage_list_last_updated	The time stamp indicating the last time that the section that is represented by the values of the executable_id and mon_interval_id monitor elements was updated.
usage_list_mem_size	The total size of the memory that is allocated for a list, in kilobytes.
usage_list_name	The usage list name.
usage_list_num_references	The total number of times that a section referenced an object since it was added to the list.
usage_list_num_ref_with_metrics	The total number of times that a section referenced an object since it was added to the list with statistics being updated.
usage_list_schema	The name of the schema of the usage list.
usage_list_size	The maximum number of entries that a usage list can hold.
usage_list_state	The status of a usage list.
usage_list_used_entries	The number of entries that were added to a usage list. If the state is I, this monitor element represents the number of entries that were captured previously when this list was activated for monitoring.
usage_list_wrapped	The value that indicates whether the list has wrapped.

The following table lists new monitor elements that report on memory allocation.

Table 16. New monitor elements for memory sets and memory pools

Name	Description
memory_pool_id	The memory pool identifier.
memory_pool_type	The name of the memory pool to identify its type.

Table 16. New monitor elements for memory sets and memory pools (continued)

Name	Description
memory_pool_used_hwm	The highest amount of memory, in KB, that was assigned to this pool since it was created.
memory_set_committed	The amount of memory, in KB, that is currently committed to a memory set.
memory_set_id	The numeric identifier for a specific memory set type.
memory_set_size	The memory commitment limit, in KB.
memory_set_type	The type of memory set.
memory_set_used_hwm	The highest amount of memory, in KB, that was assigned to memory pools from a set since the memory set was created.
memory_set_used	The amount of memory, in KB, from a set that was assigned to memory pools.

The following table lists new miscellaneous monitor elements.

Table 17. Miscellaneous new monitor elements

Name	Description
disabled_peds	The number of times that partial early distinct operations were disabled because insufficient sort heap was available.
edu_ID	The ID of the EDU with which a memory pool is associated.
evmon_waits_total	The number of times that an agent waited for an event monitor record to become available. See also <code>evmon_wait_time</code> .
index_jump_scans	The number of jump scans. A jump scan is an index scan where there is a gap in the index start and stop keys and sections of the index that will not yield results are skipped.
index_name	The name of an index.
index_schema	The name of an index schema.
mon_interval_id	The value of the <code>MON_INTERVAL_ID</code> global variable at the time that a transaction was completed.
num_page_dict_built	The number of page-level compression dictionaries that were created or re-created for a table.
post_threshold_peds	The number of times that partial early distinct operations received less memory than requested because the sort heap threshold was exceeded.
total_peas	The total number of times that partial early aggregation operations were executed.
total_peds	The total number of times that partial early distinct operations were executed.
total_stats_fabrications	The total number of statistics fabrications that real-time statistics gathering performed. See also <code>total_stats_fabrication_time</code> .
total_sync_runstats	The total number of synchronous <code>RUNSTATS</code> command activities that real-time statistics gathering triggered. See also <code>total_sync_runstats_time</code> .

Table 17. Miscellaneous new monitor elements (continued)

Name	Description
tq_sort_heap_rejections	The number of times that table queues requested for sort heap memory and was rejected because the sort heap threshold was exceeded.
tq_sort_heap_requests	The number of times that table queues requested sort heap memory to store data.

Related concepts:

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

"Monitor elements" in Database Monitoring Guide and Reference

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

Version 10.1 includes enhancements that help ensure that your data remains available.

The following enhancements are included:

- HADR now supports multiple standby databases (see “HADR now supports multiple standby databases”)
- High-speed ingest utility allows movement of real-time data without affecting availability (see “High-speed ingest utility allows movement of real-time data without affecting availability” on page 48)
- Control when an automatic failback occurs in a DB2 pureScale environment (see “**db2cluster** command now supports repairing an instance domain and controlling when an automatic failback occurs” on page 91)
- HADR time-delayed replay provides protection against application error (see “HADR delayed replay provides protection against application errors” on page 49)
- HADR log spooling prevents spikes on throughput (see “HADR log spooling prevents spikes on throughput” on page 49)
- Schema-level replication is now supported (see “Replication improvements” on page 50)

HADR now supports multiple standby databases

The high availability disaster recovery (HADR) feature now allows up to three HADR standby databases. A multiple standby setup improves your ability to protect your data while still keeping it highly available, all with a single technology.

In previous releases, the HADR feature allowed only a single standby database, meaning that you could have your data in at most two sites. With multiple standbys, you can guard against a scenario in which a region-wide outage or disaster brings down both the primary and the standby databases. For example, you can have your primary and one of your standby databases in the same location, with one or two additional standbys a long distance away. These distant standbys automatically run in SUPERASYNC mode, so the distance does not have an impact on activity on the primary database.

Another benefit of having multiple standbys is that they eliminate the implicit trade-off between high availability and disaster recovery. You can have one standby database, the *principal HADR standby database*, meet your high availability requirements by configuring it to run in close synchronization with the primary and by setting up this standby database for timely, automated failover in the event of an outage. You can also have one or two other standby databases, the *auxiliary HADR standby databases*, meet your disaster recovery requirements by situating them in a remote site. Previously, the only way to achieve this kind of setup was to use HADR for the first requirement and a different technology for the second.

All the standby databases support the HADR reads on standby feature, and all of them support both forced and unforced takeovers. In addition, you can use one of

the standbys with the new time-delayed replay feature. Using this feature, you can keep a standby behind the primary in terms of log replay so that you have time to recover from application errors that cause data loss on the primary.

Related concepts:

"HADR multiple standby databases " in Data Recovery and High Availability Guide and Reference

High-speed ingest utility allows movement of real-time data without affecting availability

The ingest utility is a new client-side DB2 utility specializing in the high speed, continuous ingestion of data from sources like files and pipes into DB2 target tables and for frequently populating data warehouse tables with minimal impact on concurrent user workload and data server resources.

The ingest utility allows you to roll-in business-critical data, even while long-running queries are accessing the table. In other words, there is no trade-off between data concurrency and data availability. The ingest utility works by continuously pumping data into DB2 tables using SQL array inserts, updates, and deletes until sources are exhausted. All ingest operations are, by default, restartable in the event of a failure. As with the load utility, users have the option of either restarting or terminating failed ingest operations.

Unlike some batch loaders that support only a few basic SQL statements, the `INGEST` command supports a variety of SQL operations, including insert, update, merge, replace, and delete. In addition, you can use SQL expressions to build individual column values from more than one data field.

The following is a list of new commands for the ingest utility:

- **INGEST**
- **INGEST SET**
- **INGEST GET STATS**
- **INGEST LIST**

The following is a list of configuration parameters for the ingest utility:

- **commit_count** - Commit count
- **commit_period** - Commit period
- **num_flushers_per_partition** - Number of flushers per partition
- **num_formatters** - Number of formatters
- **pipe_timeout** - Pipe timeout
- **retry_count** - Retry count
- **retry_period** - Retry period
- **shm_max_size** - Maximum size of shared memory

Related concepts:

"Ingest utility" in Data Movement Utilities Guide and Reference

Storing archived log files requires less disk space

To reduce the amount of disk space required to store archived log files, archived log files can now be compressed when stored.

The main benefit of this solution results in reduced storage costs associated with roll-forward recoverable databases. DB2 for Linux, UNIX, and Windows has existing mechanisms to compress data and indexes in the actual database, as well as to compress backup images. This solution adds the capability to compress archived log files. Archived log files are the third major space consumer for roll-forward recoverable databases.

Archived log files contain a considerable amount of data. They can grow quickly, especially for highly concurrent OLTP scenarios. If the modified data is in compressed tables, logging disk space is already reduced by virtue of including compressed record images in the log records. However, there is potential for further storage cost savings if compression is applied to the archived log files themselves.

This feature is available in all DB2 for Linux, UNIX, and Windows editions that support backup compression. Like backup compression, this new function does not require a DB2 Storage Optimization Feature license in DB2 Enterprise Server Edition.

Related concepts:

"Archived log file compression" in Data Recovery and High Availability Guide and Reference

HADR delayed replay provides protection against application errors

You can use a new database configuration parameter, `hadr_replay_delay`, to help protect your data from application errors.

You can use the `hadr_replay_delay` configuration parameter on your high availability disaster recovery (HADR) standby database to specify a delay for replaying logs and applying changes to the standby. By intentionally having a standby at an earlier point in time than the HADR primary, you can avoid data loss caused by errant transactions. If one of these transactions occurs on the primary, you can recover that data from the standby if you detect the problem before the replay delay has elapsed.

Related concepts:

"HADR delayed replay" in Data Recovery and High Availability Guide and Reference

Related reference:

"`hadr_replay_delay` - HADR replay delay configuration parameter" in Database Administration Concepts and Configuration Reference

HADR log spooling prevents spikes on throughput

HADR log spooling is a new feature that allows you to specify additional space where logs can be spooled on the standby. This helps avoid back-pressure issues on the primary caused by sudden spikes in logging activity on the standby.

You enable log spooling by using the `hadr_spool_limit` database configuration parameter, which sets an upper limit on how much data is written, or *spooled*, to disk if the log receive buffer fills up. Log replay on standby can later on read the log data from disk.

This feature allows transactions on HADR primary to make progress without having to wait for the log replay on HADR standby. It is aimed at reducing

primary database impact caused by slow log replay on the standby. You might experience slow replay if there is a sudden spike in transaction volume on the primary or when intensive operations like reorgs are replayed on the standby. Log spooling does not compromise the high availability (HA) and disaster recovery (DR) protection provided by HADR. The data shipped from the primary data is still replicated to standby using the specified synchronization mode; it just takes time to replay the data to the table spaces on the standby.

Related concepts:

"HADR log spooling" in Data Recovery and High Availability Guide and Reference

Related reference:

"`hadr_spool_limit` – HADR log pool limit configuration parameter" in Database Administration Concepts and Configuration Reference

Replication improvements

Starting in Version 10.1, replication is now supported at the schema level. This means that any newly created table is set up for replication.

In previous releases, you had to use the CREATE TABLE statement to enable replication. Now, by using the DATA CAPTURE attribute with the CREATE SCHEMA statement or by setting the `dft_schemas_dcc` database configuration parameter to ON, you can have all subsequently created tables inherit the DATA CAPTURE CHANGES property.

Related reference:

"CREATE SCHEMA " in SQL Reference Volume 2

"`dft_schemas_dcc` - Default data capture on new schemas configuration parameter" in Database Administration Concepts and Configuration Reference

Chapter 7. Performance enhancements

Version 10.1 contains numerous SQL 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 SQL query optimizer has been improved with the following enhancements:

- Enhanced query performance for common SQL statements (see “Enhanced query performance for common SQL statements”)
- RUNSTATS and database statistics improvements (see “RUNSTATS and database statistics improvements” on page 53)
- Optimization profile supports registry variables and inexact matching (see “Optimization profile supports registry variables and inexact matching” on page 54)
- Statistical views enhance statistics and statistic gathering for the query optimizer (see “Statistical views enhance statistics and statistic gathering for the query optimizer” on page 54)
- Intra-partition parallelism improvements (see “Intra-partition parallelism improvements” on page 55)
- Enhanced memory sharing on large POWER7[®] systems running AIX[®] (see “Enhanced memory sharing on large POWER7 systems running AIX” on page 57)
- Improved query performance through more efficient data and index prefetching (see “Improved query performance through more efficient data and index prefetching” on page 57)
- Improved performance for queries on tables with composite indexes (see “Improved performance for queries on tables with composite indexes” on page 58)
- Improved performance of star schema based queries (see “Performance of star schema based queries is improved” on page 60)

Enhanced query performance for common SQL statements

A number of performance improvements have been included in DB2 Version 10.1 to improve the speed of many queries.

These improvements are automatic; there are no configuration settings or changes to the SQL statements required.

Partial early distinct (PED)

An efficient hashing function will now be used to partially remove duplicates early in the processing of the query. This may not remove all duplicates, but will reduce the amount of data that must be processed later in the query evaluation. Removing some of the initial duplicate rows will speed up the query, and reduce the chance that it will run out of sort heap memory, thereby eliminating the need to use relatively slow disk space for temporary storage in these cases. This improvement is termed partial early distinct (PED).

To determine if this improvement is being used for a particular query, activate the Explain facility and run the query. A new value in the EXPLAIN_ARGUMENT table indicates when this new functionality has been applied to a query:

- ARGUMENT_TYPE column = UNIQUE
- ARGUMENT_VALUE column can now also have the value: HASHED PARTIAL which indicates that the new feature has been used

The **db2exfmt** tool will also show HASHED PARTIAL in its output, as shown in the following example:

```
6) UNIQUE: (Unique)
   Cumulative Total Cost:   132.519
   Cumulative CPU Cost:    1.98997e+06
   ...
   ...
   Arguments:
   -----
   JN INPUT: (Join input leg)
           INNER
   UNIKEY : (Unique Key columns)
           1: Q1.C22
   UNIKEY : (Unique Key columns)
           2: Q1.C21
   pUNIQUE : (Uniqueness required flag)
           HASHED PARTIAL
```

Partial early aggregation (PEA)

Similar to partial early distinct (PED), partial early aggregation (PEA) is an attempt to do a partial aggregation of data early in the processing of the query. While it is unlikely that all aggregation can take place at this point, it will at least reduce the amount of data that must be processed later in the query evaluation.

To determine if partial early aggregation is being used for a particular query activate the Explain facility and run the query. A new value in the EXPLAIN_ARGUMENT table indicates when this new functionality has been applied to a query:

- ARGUMENT_TYPE column = AGGMODE
- ARGUMENT_VALUE column can now also have the value: HASHED PARTIAL which indicates that this new feature has been used

The **db2exfmt** tool will also show HASHED PARTIAL in its output for GRPBY sections, along with a pGRPBY in the tree view, if this new functionality has been applied within that part of the query.

Hash join now selected by the query optimizer for a wider range of SQL queries

The query optimizer chooses between three basic join strategies when determining how to run an SQL query that includes a join. In many cases a hash join is the most efficient method, and with this release it can be used in more situations.

Data type mismatches

A hash join will now be considered even if the two columns in the join are not the same data type. This is the case in all but the most extreme situations.

Expressions used in join predicate

Join predicates that contain an expression no longer restrict the join

method to a nested loop join. In this release a hash join is considered in cases where the WHERE clause contains an expression, like: WHERE T1.C1 = UPPER(T1.C3)

In these cases the hash join is considered automatically. There is no need to change any existing SQL queries to take advantage of this improved functionality. Note that hash joins make use of sort heap memory.

Improved cost estimates of network communication traffic generated by a query

The query optimizer relies on a range of information to choose an access plan that is efficient as possible. The estimated communication costs of queries has now been improved, enabling the optimizer to more accurately consider and compare all of the CPU, IO, and communication costs. In many cases this will result in faster query performance.

The estimated per-node communication costs of a query, as returned by the explain elements **COMM_COST** and **FIRST_COMM_COST**, have been improved. They are now more consistent with the existing CPU and IO costs per-node calculations. This enables the query optimizer to effectively balance all three of these cost estimations when evaluating different access plans. It also helps increase parallelism when possible by enabling the network traffic to be spread more evenly across multiple network adapters. In particular:

- If there is more than one network adapter involved, the cumulative communication cost for the adapter with the highest value is returned. In previous releases the total number of frames transmitted throughout the entire network was returned.
- The values only include the costs of network traffic between physical machines. They do not include the virtual communication costs between node partitions on the same physical machine in a partitioned database environment.

RUNSTATS and database statistics improvements

A number of improvements have been made to the **RUNSTATS** command to make statistics gathering faster in some cases. The command parameters have also been simplified.

RUNSTATS now supports index sampling

The **RUNSTATS** command can now collect index statistics using a sampling method rather than by scanning the entire index. This is activated with a new command parameter: **INDEXSAMPLE**. The interface is similar to the existing **TABLESAMPLE** command parameter. The new sampling will usually speed up the time it takes to generate statistics by either reducing the total number of leaf nodes processed by **RUNSTATS** (when **INDEXSAMPLE SYSTEM** is specified) or by reducing the total number of index entries processed by **RUNSTATS** (when **INDEXSAMPLE BERNOULLI** is specified).

Starting from DB2 Version 10.1, the default method of gathering detailed index statistics has changed. When the **DETAILED** option is used, it no longer scans the entire index but rather uses a sampling method to gather statistics. This option is now equivalent to the **SAMPLED DETAILED** option which has been left in for compatibility. To gather detailed index statistics by scanning the entire index, as in previous releases, you can specify the **UNSAMPLED** option.

RUNSTATS command improvements

New VIEW command parameter

The **RUNSTATS** command now supports the **VIEW** command parameter. This parameter has been added for more intuitive usability when running **RUNSTATS** on views. The command runs the same way as if the parameter **TABLE** was specified for views.

Schema name specification

To improve the usability of the **RUNSTATS** command, you no longer have to fully qualify object names by specifying a schema name. If you do not specify a schema name, the default schema is used.

Automatic sampling possible for all background statistic collection

It is now possible to enable automatic sampling for all background statistics collection on large tables and statistical views. Often the statistics generated by considering just a sampling of the data is as accurate as considering the entire table or view, but can take less time and resources to complete. Use the new `auto_sampling` parameter to enable this.

Related reference:

"**RUNSTATS** " in Command Reference

Optimization profile supports registry variables and inexact matching

The optimization profile can now be used to set certain registry variables and supports inexact matching. Inexact matching can be used for better matching when compiling query statements.

A subset of registry variables can be set in the optimization profile with the **OPTION** element in the **REGISTRY** element. The **OPTION** element has **NAME** and **VALUE** attributes where you specify the registry variable and its value. You can set many registry variables at the global level or, for specific statements, at the statement-level.

The optimization profile now supports inexact matching in addition to exact matching. Inexact matching ignores literals, host variables, and parameter markers when statements are being matched. To specify inexact matching in the optimization profile, set the **EXACT** attribute value of the **STMTMATCH** element to **FALSE**. You can specify the **STMTMATCH** element at both the global level or at the statement-level.

Related concepts:

"Statement key and compilation key matching" in Troubleshooting and Tuning Database Performance

"SQL compiler registry variables in an optimization profile" in Troubleshooting and Tuning Database Performance

Statistical views enhance statistics and statistic gathering for the query optimizer

There are new features of statistical views which the DB2 query optimizer can now use to generate better access plans, improving the performance of certain queries.

Predicates containing complex expressions

The DB2 query optimizer can now use statistics from expression columns, a column with one or more functions, in statistical views. In previous releases, the optimizer could use only a default value for the selectivity estimate for queries that have complex expressions in the predicate. Starting in this release, however, the optimizer can now use the actual statistics to generate better access plans.

Reducing the number of statistical views

The number of statistical views required to obtain good statistics for a star join query can now be reduced if referential integrity constraints exist and are defined in the data. You can now create one statistical view which contains many of the columns from the join queries. The statistics for specific joins are inferred from this one statistical view based on referential integrity constraints.

Column group statistics gathered on statistical views

Statistics can now be used by the DB2 query optimizer from statistical views where column group statistics are collected on the views. Combining column group statistics with statistical views improves access plans because the optimizer can use the adjusted statistics gathered from queries that might be skewed.

Automatic statistics collection for statistical views

The DB2 automatic statistics collection function can now automatically gather statistics for statistical views. This function is not enabled by default and must be turned on using a new database configuration parameter, `auto_stats_view`. This new parameter must be turned on using the `UPDATE` command in order for statistics to be automatically gathered from statistical views. The statistics collected by the automatic statistics collection is equivalent to issuing the following command: `runstats on view <view_name> with distribution.`

Related concepts:

"Collecting accurate catalog statistics, including advanced statistics features" in Troubleshooting and Tuning Database Performance

"Statistics used from expression columns in statistical views" in Troubleshooting and Tuning Database Performance

"Referential integrity constraints help reduce the number of statistical views" in Troubleshooting and Tuning Database Performance

"Statistics used from column group statistics on statistical views" in Troubleshooting and Tuning Database Performance

Intra-partition parallelism improvements

One goal of the DB2 query optimizer is to choose parallel execution strategies that maintain data balance among subagents and keep them equally busy. In this release, the parallelization capabilities of the optimizer have been further enhanced to enable more workloads to better use multi-core processors.

Rebalancing imbalanced subagent workloads

Data filtering and data skew can cause workloads between subagents to become imbalanced while a query executes. The inefficiency of imbalanced workloads is magnified by joins and other computationally expensive operations. The optimizer looks for sources of imbalance in the query's access plan and applies a balancing

strategy, ensuring that work is evenly divided between the subagents. For an unordered outer data stream, the optimizer balances the join using the REBAL operator on the outer. For an ordered data stream (where ordered data is produced by an index access or a sort), the optimizer balances the data using a shared sort. A shared sort will not be used if the sort overflows into the temporary tables, due to the high cost of a sort overflow.

Parallel scans on range partitioned tables and indexes

Parallel table scans can be run against range partitioned tables, and similarly, parallel index scans can be run against partitioned indexes. For a parallel scan, partitioned indexes are divided into ranges of records, based on index key values and the number of key entries for a key value. When a parallel scan begins, subagents are assigned a range of records, and once the subagent completes a range, it is assigned a new range. The index partitions are scanned sequentially with subagents potentially scanning unreserved index partitions at any point in time without waiting for each other. Only the subset of index partitions that is relevant to the query based on data partition elimination analysis is scanned.

Ability to throttle the degree of parallelism to optimize for transactional workloads

Individual applications or workloads can now dynamically throttle the degree of intra-partition parallelism to optimize performance for the types of queries being executed. In previous versions of DB2, it was only possible to control the degree of parallelism (and whether it was turned on or off) for the whole instance. Turning parallelism on or off also required the instance to be restarted. On database servers with mixed workloads, a more flexible approach to controlling intra-partition parallelism is needed. Transactional workloads, which typically include short insert, update, and delete transactions, do not benefit from parallelization. There is some processing overhead when intra-partition parallelism is enabled, which introduces a negative impact to transactional workloads. However, data warehouse workloads benefit greatly from parallelization as they typically include processor-intensive long-running queries.

For mixed workloads, with transactional and data warehousing components, you can now configure the database system to provide parallelism settings that are optimal for the kind of workload deployed by each application. You can either control the parallelism settings through application logic, or through the DB2 workload manager (which does not require application changes).

Controlling intra-partition parallelism from database applications: To enable or disable intra-partition parallelism from within a database application, you can call the new `ADMIN_SET_INTRA_PARALLEL` procedure. For example, the following statement enables intra-partition parallelism:

```
CALL ADMIN_SET_INTRA_PARALLEL('YES')
```

Although the procedure is called in the current transaction, it takes effect starting with the following transaction, and is only applicable to the calling application. The setting for intra-partition parallelism set by `ADMIN_SET_INTRA_PARALLEL` will override whatever value is in the `intra_parallel` configuration parameter.

Controlling intra-partition parallelism from the DB2 workload manager: To enable or disable intra-partition parallelism for a specified workload, you can set the `MAXIMUM_DEGREE` workload attribute. For example, the following statement disables intra-partition parallelism for a workload called `trans`:

ALTER WORKLOAD trans MAXIMUM DEGREE 1

All statements in the workload executed after the ALTER WORKLOAD statement will be run with intra-partition parallelism turned off. The setting for intra-partition parallelism set with the MAXIMUM DEGREE workload attribute overrides calls to ADMIN_SET_INTRA_PARALLEL, and will override whatever value is in the `intra_parallel` configuration parameter.

Related concepts:

"Optimization strategies for intra-partition parallelism" in Troubleshooting and Tuning Database Performance

Enhanced memory sharing on large POWER7 systems running AIX

A DB2 Version 10.1 system can now determine the hardware topology on AIX systems running POWER7 to potentially enhance query performance by effectively sharing memory.

If you set the `DB2_RESOURCE_POLICY` variable to `AUTOMATIC` then the DB2 database system will automatically determine the hardware topology and assign engine dispatchable units (EDUs) to the various hardware modules in such a way that memory can be more efficiently shared between multiple EDUs that need to access the same regions of memory.

This setting is intended to be used on larger POWER7 systems with 16 or more cores and can result in enhanced query performance on some workloads. It is best to run a performance analysis of the workload before and after setting this variable to `AUTOMATIC` to validate any performance improvements.

Related reference:

"Performance variables" in Database Administration Concepts and Configuration Reference

Improved query performance through more efficient data and index prefetching

DB2 Version 10.1 features *smart data prefetching* and *smart index prefetching*, which improves query performance and reduces the need to reorganize tables and indexes.

After many changes to table data or indexes, sequential data or indexes might reside on badly clustered data pages or low density index leaf pages. In previous releases this would likely result in degraded query performance. This is because sequential detection prefetching becomes less efficient as badly clustered data pages increases and as the density of index leaf pages lowers.

Smart data prefetching is applied only during an `ISCAN-FETCH`, and smart index prefetching is applied during any index scan even if it part of an `ISCAN-FETCH`. The optimizer can combine smart data and smart index prefetching to select an optimal index and data prefetching technique. However, smart index and smart data prefetching are independent of each other.

A new type of prefetching introduced in DB2 Version 10.1 called *readahead prefetching* is used to efficiently prefetch badly clustered data pages and low density index pages. Except in the following case of the restrictions, the optimizer selects readahead prefetching as a backup to sequential detection prefetching. At

runtime, the prefetching type might switch from sequential detection prefetching to readahead prefetching when it detects that sequential detection prefetching is not working well enough. Readahead prefetching looks ahead in the index to determine the exact data pages or index leaf pages that an index scan operation will access, and prefetches them. While readahead prefetching provides all the data pages and index leaf pages needed during the index scan (and no pages that are not needed), it also requires additional resources to locate those pages. For highly sequential data or indexes, sequential detection prefetching will most often out-perform readahead prefetching.

Smart data prefetching is an approach where either sequential detection or readahead prefetching is used, depending on the degree of data clustering. Sequential detection prefetching is used when the data pages are stored sequentially, and readahead prefetching is used when the data pages are badly clustered. Smart data prefetching enables the database system to capitalize on the potential performance benefits of data stored in sequential pages, while also enabling badly clustered data to be prefetched efficiently. Since badly clustered data is no longer as detrimental to query performance, this reduces the need for an expensive operation like a table reorganization.

Smart index prefetching is an approach where either sequential detection or readahead prefetching is used, depending on the density of the indexes. Sequential detection prefetching is used when indexes are stored sequentially, and readahead prefetching is used when the indexes have a low density. Smart index prefetching enables the database system to capitalize on the potential performance benefits of indexes stored sequentially, while also enabling low density indexes to be prefetched efficiently. Smart index prefetching reduces the need for an expensive operation like an index reorganization.

Smart data and index prefetching support only applies to index scan operations and does not support XML, extended, and Text Search text indexes. Smart data prefetching cannot be used during scans of global range-clustered table indexes, since they are logical indexes and not physical. Also, for smart data prefetching, if the ISCAN-FETCH scans a global range partitioned index, data readahead prefetching is not used. If index predicates are evaluated during the index scan for smart data prefetching, and the optimizer determines that not many rows qualify for that index scan, readahead prefetching is disabled. Smart index prefetching also cannot be used for range-clustered table indexes.

Related concepts:

"Prefetching data into the buffer pool" in Troubleshooting and Tuning Database Performance

Improved performance for queries on tables with composite indexes

The DB2 query optimizer can now create additional access plans that might be more efficient for queries with index gaps in their start-stop keys by using a jump scan operation.

For example, index gaps are common in queries with multiple predicates that are issued against tables with composite indexes. Jump scans eliminate the need for index gap avoidance tactics such as the creation of additional indexes.

Problem: Index gaps

For workloads involving many ad-hoc queries, it is often difficult to optimize a database for high performance. Queries against tables with composite (multi-column) indexes present a particular challenge. Ideally, a query's predicates are consistent with a table's composite index. This would mean that each predicate could be used as a start-stop key, which would, in turn, reduce the scope of the index needing to be searched. When a query contains predicates that are inconsistent with a composite index, this is known as an *index gap*. As such, index gaps are a characteristic of a query, not of a table's indexes.

For example, consider a table T with integer columns A, B, and C, and a composite index defined on the columns A, B, and C. Now, consider the following query against table T:

```
SELECT * FROM t WHERE a=5 AND c=10
```

This query contains an index gap on column B in the composite index (this is assuming that the access plan contains an index scan on the composite index).

In the case of an index gap, the index scan will likely have to process many unnecessary keys. The predicates on the non-leading columns of the index would likely need to be applied individually against each key in the index that satisfies the start-stop keys. This slows down the index scan as more rows need to be processed, and additional predicates need to be evaluated for each key. Also, DB2 must sequentially examine all keys in what could be a large range.

To avoid index gaps you can define additional indexes to cover the permutations of query predicates likely to appear in your workloads. This is not an ideal solution, because defining additional indexes requires additional database administration and consumes storage capacity. Also, for workloads with many ad-hoc queries, it can be difficult to anticipate which indexes would be needed.

Solution: jump scan enablement

In DB2 Version 10.1, the query optimizer can build an access plan that uses the *jump scan* operation when queries contain index gaps. In a jump scan operation, the index manager identifies qualifying keys for small sections of a composite index where there are gaps, and fills these gaps with these qualifying keys. The result is that the index manager skips over parts of the index that will not yield any results.

Note: When evaluating queries, there can be cases where the query optimizer builds an access plan that does not include a jump scan operation, even if index gaps are present. This would occur if the query optimizer deems an alternative to using a jump scan to be more efficient.

Related concepts:

"Data access through index scans" in Troubleshooting and Tuning Database Performance

Related reference:

"index_jump_scans - Index jump scans monitor element " in Database Monitoring Guide and Reference

Performance of star schema based queries is improved

Star schema performance improvements include an improved detection algorithm and a new join method.

The improved star schema detection algorithm allows the query optimizer to detect queries based on star schema and employ star schema specific strategies to improve performance for those queries. In addition, to improve performance of queries using the star schema in data warehouse and data mart environments, the new zigzag join method can be used to join one or more fact tables with two or more dimension tables.

Improved star schema detection

The new and improved star schema detection algorithm does not base its analysis on the sizes of the tables to decide whether the query is based on star schema. Instead, it relies on the primary keys, unique indexes, or unique constraints on the dimension/snowflake tables and the join predicates between the dimension/snowflake table and the fact table. The enhanced star detection algorithm can recognize multiple stars in a query block. It removes some of the restrictions imposed by the star detection algorithm used before DB2 Database for Linux, UNIX, and Windows Version 10.1. If the new detection method cannot detect whether a query is based on a star schema, for example, if there is no primary key, unique index, or unique constraint on the dimension table, then the original detection method is used instead.

Using the jump scan feature, the query optimizer can recognize star schemas even when a join predicate is missing from the query.

New zigzag join method

Prior to this release of DB2 Database for Linux, UNIX, and Windows, there were two specific strategies to process star schema join queries:

- A Cartesian-hub join plan which computes the Cartesian product of dimensions; each row in the Cartesian product is then used to probe the multicolumn fact table index.
- A star join plan which pre-filters the fact table by dimensions to generate semi-joins, index ANDs the results of the semi-joins, and then completes the semi-joins.

In addition to these two special star join processing techniques, the new zigzag join method can now be used to expedite the processing of queries based on star schema.

A zigzag join is a join method in which a fact table and two or more dimension tables in a star schema are joined, such that the fact table is accessed using an index. It requires equality predicates between each dimension table and the fact table. This join method calculates the Cartesian product of rows from the

dimension tables without actually materializing the Cartesian product and probes the fact table using a multicolumn index, so that the fact table is filtered along two or more dimension tables simultaneously. The probe into the fact table finds matching rows. The zigzag join then returns the next combination of values that is available from the fact table index. This next combination of values, known as feedback, is used to skip over probe values provided by the Cartesian product of dimension tables that will not find a match in the fact table. Filtering the fact table on two or more dimension tables simultaneously, and skipping probes that are known to be unproductive, together makes the zigzag join an efficient method for querying large fact tables.

Related concepts:

"Ensuring that queries fit the required criteria for the star schema join" in *Troubleshooting and Tuning Database Performance*

Chapter 8. SQL compatibility enhancements

If you work with relational database products other than the DB2 product, Version 10.1 contains enhancements that make the DB2 product more familiar to you. These enhancements reduce the time and complexity of enabling some applications that are written for other relational database products to run in a DB2 environment.

The following enhancements are included:

- CREATE TRIGGER statement allows more flexibility (see “Expanded support for triggers”)
- Declared user-defined data types and procedures enhanced (see “Declared types and procedures” on page 64)
- New scalar functions increase DB2 compatibility (see “New scalar functions” on page 64)

Expanded support for triggers

A trigger defines a set of actions that are performed in response to an event like an insert, update, or delete operation on a table. Starting in Version 10.1 the CREATE TRIGGER statement allows more flexibility and functionality when creating triggers.

Multiple-event trigger support

The trigger event clause in the CREATE TRIGGER statement can now contain more than one operation. The ability to use UPDATE, DELETE, and INSERT operations together in a single clause 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, a trigger event cannot be specified more than once.

Trigger event predicates identify trigger events

The trigger event predicates of UPDATING, INSERTING, and DELETING can be used to identify the event that activated a trigger. Trigger event predicates can only be used in the trigger action of a CREATE TRIGGER statement that uses a compound SQL (compiled) statement.

FOR EACH STATEMENT restriction removed

The FOR EACH STATEMENT option is now supported in the CREATE TRIGGER statement for PL/SQL triggers. You can create triggers that fire only one time per statement irrespective of the number of rows affected.

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

Declared types and procedures

Starting in Version 10.1 you can declare user-defined data types and procedures that are local to a compound SQL (compiled) statement

Information about declared data types and procedures is not stored in the DB2 catalog. Data types can be used and procedures invoked only in the compound SQL (compiled) statements where they are declared or in nested compound SQL (compiled) statements.

Related concepts:

"Anonymous block statement (PL/SQL)" in SQL Procedural Languages: Application Enablement and Support

New scalar functions

New scalar functions are added in DB2 Version 10.1 to increase DB2 compatibility.

INSTRB

The INSTRB function returns the starting position, in bytes, of a string within another string.

TO_SINGLE_BYTE

The TO_SINGLE_BYTE function returns a string in which multi-byte characters are converted to the equivalent single-byte character where an equivalent character exists.

TIMESTAMPDIFF

The TIMESTAMPDIFF function returns an estimated number of intervals of the type defined by the first argument, based on the difference between two timestamps

Related reference:

"TIMESTAMPDIFF " in SQL Reference Volume 1

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

"INSTRB " in SQL Reference Volume 1

"TO_SINGLE_BYTE " in SQL Reference Volume 1

Chapter 9. Workload management enhancements

Version 10.1 features extend the workload management capabilities provided in previous releases.

You can now manage your workloads with DB2 workload manager in a DB2 pureScale environment:

- DB2 workload manager is now available in the DB2 pureScale environment (see “DB2 workload manager is now available in the DB2 pureScale environment” on page 69)

Management of CPU resources at the service-class level has been provided for the DB2 workload manager by the following Version 10.1 functionality:

- DB2 WLM dispatcher manages CPU resource allocations for workloads running in service classes (see “DB2 WLM dispatcher manages service class CPU resource allocations”)

Using DB2 WLM, you can now classify and prioritize activities based on which data the activity accesses:

- Using DB2 WLM, you can now classify and prioritize activities based on which data the activity accesses (see “DB2 WLM can prioritize activities based on the accessed data” on page 68)

You can now create thresholds for specific statements with the new STATEMENT threshold domain:

- You can now create thresholds for specific statements with the new STATEMENT threshold domain (see “New STATEMENT domain for thresholds lets you define thresholds for statements that contain specific text” on page 26)

DB2 WLM dispatcher manages service class CPU resource allocations

The DB2 workload manager (WLM) dispatcher is a built-in DB2 technology by which you can specifically allocate CPU resources to work that is being executed on a database server. CPU resource entitlements can be controlled by using CPU shares and CPU limit attributes on DB2 WLM user and maintenance service class objects.

In summary, the DB2 WLM dispatcher has the following benefits:

- Easy to implement, requiring less time and effort than implementing an OS WLM such as AIX WLM or Linux WLM.
- Supports flexible CPU allocation throughout the normal daily ebb and flow of system use. This flexibility is accomplished through the provisioning of both permanent allocations that are enforced all the time (hard CPU shares and CPU limits), or dynamic allocations that are enforced only when demand exceeds capacity (soft CPU shares).
- Self-contained within the DB2 database manager such that setting CPU resource entitlements provides you with workload control that is effective across all platforms due to its independence from an OS WLM such as AIX WLM or Linux WLM.
- You can continue to use the OS WLM products as a workload control mechanism, but it is not necessary if the extra implementation complexity (for

example, setting up AIX WLM on each partition) or if the organizational strife (for example, system administrator reluctance to implement or permit use of OS WLM) is a barrier. Alternatively, you can use the OS WLM products for monitoring purposes while relying on the DB2 WLM dispatcher for workload control.

You can use the DB2 WLM dispatcher to effectively manage DB2 workloads by allocating CPU resource entitlements, that you specify, without the need for third-party workload manager software. The dispatcher can control CPU resource entitlements for DB2 workloads by means of CPU shares-based settings and CPU limit settings. Uncapped soft CPU shares provide almost unrestricted use of unused CPU resources, whenever they become available, that you assign to your high-priority work. You can assign capped hard CPU shares or CPU limits to your low-priority work. Hard CPU shares and CPU limits are most effective in preventing low-priority work from interrupting the execution of your high-priority work. Yet, low-priority work, to which hard CPU shares are usually assigned, has the flexibility to consume unused CPU resources that have been relinquished by high-priority work that has become idle or fallen below a minimum CPU utilization level to be considered active; this scenario can typically occur during off-peak business hours. Hard CPU shares and CPU limits are most useful in environments where the CPU utilization is typically low, without the need to assign soft CPU shares which are most effective in environments where the CPU utilization is almost always high.

The dispatcher infrastructure operates at the instance level of the DB2 database manager. The WLM dispatcher determines which DB2 agents can run based on the CPU allocation for their service class.

To enable the WLM dispatcher, you must set the **wlm_dispatcher** database manager configuration parameter to YES (by default, this configuration parameter is set to NO). By default after the WLM dispatcher has been enabled, the dispatcher can manage CPU resources only by way of CPU limit settings.

After having made the decision that the earlier enabled WLM dispatcher can best manage your strained CPU resources by using CPU shares along with CPU limits, you must enable CPU shares by setting the **wlm_disp_cpu_shares** database manager configuration parameter to YES. The default setting for this parameter is NO. You can set and adjust CPU shares and CPU limits by using the CREATE SERVICE CLASS and ALTER SERVICE CLASS statements.

Another consideration that gives you maximum flexibility to control the behavior of the DB2 database manager is the option to set a minimum percentage of CPU resource utilization for service classes using the **wlm_disp_min_util** database manager configuration parameter. Service classes utilizing CPU resources equal to or greater than that minimum percentage are considered to be active on the host or logical partition (LPAR) and the CPU shares of the active service classes are factored into CPU resource entitlement calculations.

The WLM dispatcher can manage a number of simultaneous threads that is referred to as the dispatch concurrency level. You can set the dispatch concurrency level using the **wlm_disp_concur** database manager configuration parameter. You can either specify that the DB2 database manager itself sets the concurrency level (COMPUTED), or you can manually set the concurrency level to a fixed value.

Workload monitoring has been enhanced to support the WLM dispatcher technology. New and enhanced monitor elements and table functions are as follows:

- New monitor elements:
 - act_throughput - Activity throughput monitor element
 - cpu_limit - WLM dispatcher CPU limit monitor element
 - cpu_share_type - WLM dispatcher CPU share type monitor element
 - cpu_shares - WLM dispatcher CPU shares monitor element
 - cpu_utilization - CPU utilization monitor element
 - cpu_velocity - WLM dispatcher CPU velocity monitor element
 - estimated_cpu_entitlement - Estimated CPU entitlement monitor element
 - total_disp_run_queue_time - Total dispatcher run queue time monitor element
 - uow_completed_total - Total completed units of work monitor element
 - uow_lifetime_avg - Unit of work lifetime average monitor element
 - uow_throughput - Unit of work throughput monitor element
- Enhanced monitor elements:
 - db_name - Database name monitor element
 - histogram_type - Histogram type monitor element
 - hostname - Host name monitor element
 - total_cpu_time - Total CPU time monitor element
- New table functions:
 - MON_SAMPLE_SERVICE_CLASS_METRICS - Get sample service class metrics
 - MON_SAMPLE_WORKLOAD_METRICS - Get sample workload metrics
- Enhanced table functions:
 - MON_GET_ACTIVITY_DETAILS table function - Get complete activity details
 - MON_GET_CONNECTION table function - Get connection metrics
 - MON_GET_CONNECTION_DETAILS table function - Get detailed connection metrics
 - MON_GET_PKG_CACHE_STMT table function - Get SQL statement activity metrics in the package cache
 - MON_GET_PKG_CACHE_STMT_DETAILS table function - Get SQL statement activity metrics in the package cache
 - MON_GET_SERVICE_SUBCLASS table function - Get service subclass metrics
 - MON_GET_SERVICE_SUBCLASS_DETAILS table function - Get detailed service subclass metrics
 - MON_GET_UNIT_OF_WORK table function - Get unit of work metrics
 - MON_GET_UNIT_OF_WORK_DETAILS table function - Get detailed unit of work metrics
 - MON_GET_WORKLOAD table function - Get workload metrics
 - MON_GET_WORKLOAD_DETAILS table function - Get detailed workload metrics
 - WLM_GET_SERVICE_SUBCLASS_STATS table function - Get service subclass statistics
 - WLM_GET_WORKLOAD_STATS table function - Get workload statistics

Related concepts:

"Workload management dispatcher overview" in DB2 Workload Management Guide and Reference

DB2 WLM can prioritize activities based on the accessed data

Using DB2 WLM, you can now prioritize an activity based on the data that the activity accesses, either before the activity executes (predictively) or while the activity is executing (reactively).

To prioritize an activity, you use a combination of a *data tag*, which is a numeric identifier applied to a table space or storage group, and WLM controls. For example, if you have a table space IMPORTANT_TS containing critical data that has a data tag assigned to it, you could map any query that reads data from a table in this table space to a service class that is allocated a higher percentage of overall CPU cycles on the system.

You can assign a data tag directly to a table space or assign the data tag to the storage group for the table space and have the table space inherit the data tag from the storage group.

Predictive prioritization using work class and work action sets uses an estimated data tag list that is obtained for an activity at compile time, similar to cost and cardinality estimates. The estimated data tag list contains the data tags for all table spaces that the compiler believes will be accessed during execution of the activity. You can define work class sets to identify activities that have a particular data tag in their estimated data tag lists. You can then define a work action to map any activities matching a work class set to a specific service class before they begin to execute.

Reactive prioritization using the new DATATAGINSC threshold maps an activity to a different service class at run time when the activity accesses data that is assigned a particular data tag. For example, you can specify that an activity will be mapped to a different service class when it reads data from a table space with data tag value of 3. Reactive prioritization is useful if the compiler cannot accurately estimate the list of data tags for the activity. An example of such a case is a query against a range-partitioned table that uses parameter markers. The compiler cannot necessarily determine what table ranges are accessed in advance.

To support data tags, the following DB2 commands SQL reference statements have been added or modified:

- The output of the **-tablespace** parameter for the **db2pd** command now includes information about data tags.
- The output of the **-workclasses** parameter for the **db2pd** command now lists the work class attributes below the basic work class information.
- The ALTER TABLESPACE statement has the new DATA TAG clause.
- The ALTER THRESHOLD statement has the new DATATAGINSC clause.
- The ALTER WORK CLASS SET statement has the new DATA TAG LIST CONTAINS clause.
- The CREATE TABLESPACE statement has the new DATA TAG clause.
- The CREATE THRESHOLD statement has the new DATATAGINSC clause.
- The CREATE WORK CLASS SET statement has the new DATA TAG LIST CONTAINS clause.

Related concepts:

"Multi-temperature data storage can provide fast access to data" on page 10

Related tasks:

"Creating a work class set" in DB2 Workload Management Guide and Reference

Related reference:

"CREATE THRESHOLD " in SQL Reference Volume 2

"ALTER THRESHOLD " in SQL Reference Volume 2

"DATATAGINSC threshold" in DB2 Workload Management Guide and Reference

DB2 workload manager is now available in the DB2 pureScale environment

With DB2 Version 10.1, you are now able to use DB2 workload manager (DB2 WLM) to manage your workloads when the IBM DB2 pureScale Feature is enabled. A good workload management configuration aids in maximizing system efficiency and throughput, while helping you to achieve your business performance objectives.

Related concepts:

"Introduction to DB2 workload management concepts" in DB2 Workload Management Guide and Reference

Chapter 10. Security enhancements

With the number of both internal and external security threats growing, it is important to separate the tasks of keeping data secure from the management tasks of administering critical systems. Building on the enhancements introduced in previous versions, Version 10.1 enhancements ensure that your sensitive data is even better protected.

The following enhancements are included:

- .Data security enhancement (see “Row and column access control (RCAC) enhances data security”).

Row and column access control (RCAC) enhances data security

DB2 Version 10.1 introduces row and column access control (RCAC) as a solution to help you further secure your data. RCAC is sometimes referred to as fine-grained access control or FGAC.

Row and column access control allows you to regulate data access at the row level, column level, or both. RCAC can be used to complement the table privileges model.

You can rely upon row and column access control to ensure that your users have access to only the data that is required for their work.

Related concepts:

"Row and column access control (RCAC) overview" in Database Security Guide

Chapter 11. Application development enhancements

Version 10.1 application development enhancements simplify database application development, improve application portability, and ease application deployment.

The following enhancements are included:

- Built-in global variables expand SQL programming capabilities (see “Built-in global variables expand SQL programming capabilities”)
- Use Java generic table functions for custom analytics (see “Use Java generic table functions for custom analytics” on page 74)
- Manage and query time-based data using temporal tables (see “Manage and query time-based data using temporal tables” on page 75)
- IBM data server clients and drivers enhancements (see “IBM data server clients and drivers enhancements” on page 76)

Built-in global variables expand SQL programming capabilities

Built-in global variables are global variables that are created with the database manager and are registered to the database manager in the system catalog.

The built-in global variables introduced in DB2Version 10.1 are generated automatically during database creation. Once created, you can access the values of these global variables programmatically through SQL statements to share data without the need for additional application logic.

The following built-in global variables are now available.

CLIENT_HOST

This built-in global variable contains the host name of the current client, as returned by the operating system.

CLIENT_IPADDR

This built-in global variable contains the IP address of the current client, as returned by the operating system.

CLIENT_ORIGUSERID

This built-in global variable contains the original user identifier, as supplied by an external application such as an application server over an explicit trusted connection.

CLIENT_USRSECTOKEN

This built-in global variable contains a security token, as supplied by an external application such as an application server over an explicit trusted connection.

MON_INTERVAL_ID

This built-in global variable contains the identifier for the current monitoring interval.

PACKAGE_NAME

This built-in global variable contains the name of the currently executing package.

PACKAGE_SCHEMA

This built-in global variable contains the schema name of the currently executing package.

PACKAGE_VERSION

This built-in global variable contains the version identifier of the currently executing package.

ROUTINE_MODULE

This built-in global variable contains the module name of the currently executing routine.

ROUTINE_SCHEMA

This built-in global variable contains the schema name of the currently executing routine.

ROUTINE_SPECIFIC_NAME

This built-in global variable contains the specific name of the currently executing routine.

ROUTINE_TYPE

This built-in global variable contains the type of the currently executing routine.

TRUSTED_CONTEXT

This built-in global variable contains the name of the trusted context that was matched to establish the current trusted connection.

Related concepts:

"Built-in global variables" in SQL Reference Volume 1

Related reference:

"ROUTINE_MODULE global variable" in SQL Reference Volume 1

"ROUTINE_SCHEMA global variable" in SQL Reference Volume 1

"ROUTINE_TYPE global variable" in SQL Reference Volume 1

"ROUTINE_SPECIFIC_NAME global variable" in SQL Reference Volume 1

"CLIENT_HOST global variable" in SQL Reference Volume 1

"CLIENT_IPADDR global variable" in SQL Reference Volume 1

"CLIENT_ORIGUSERID global variable" in SQL Reference Volume 1

"CLIENT_USRSECTOKEN global variable" in SQL Reference Volume 1

"MON_INTERVAL_ID global variable" in SQL Reference Volume 1

"PACKAGE_NAME global variable" in SQL Reference Volume 1

"PACKAGE_SCHEMA global variable" in SQL Reference Volume 1

"PACKAGE_VERSION global variable" in SQL Reference Volume 1

"TRUSTED_CONTEXT global variable" in SQL Reference Volume 1

Use Java generic table functions for custom analytics

With generic table functions, you can specify the output of a table function when you reference it, rather than when you create it.

To define a generic table function, use the CREATE FUNCTION statement, and specify the RETURNS GENERIC TABLE option. To use this option, you must also specify the LANGUAGE JAVA and PARAMETER STYLE DB2GENERAL options.

After the function is defined, you can access the function output using an SQL select statement, which includes a typed-correlation clause. The typed-correlation clause defines the schema of the result table, including column names and data types. You can use different select statements to output tables of different schemas from the same generic table function.

Related reference:

"Java classes for DB2GENERAL routines" in *Developing User-defined Routines (SQL and External)*

Manage and query time-based data using temporal tables

Use temporal tables associated with Time Travel Query to assign time-based state information to your data. Data in tables that do not use temporal support represents the present, while data in temporal tables is valid for a period defined by the database system, customer applications, or both.

For example, a database can store the history of a table (deleted rows or the original values of rows that have been updated) so you can query the past state of your data. You can also assign a date range to a row of data to indicate when it is deemed to be valid by your application or business rules.

For many businesses there are important reasons to preserve the history of data changes. Without this capability in the database, it is expensive and complex for businesses to maintain audit trails for regulatory compliance.

Many businesses also need to track the time period when a data row is deemed to be valid from a business perspective. For example, the time period an insurance policy is active. There might also be a need to store future data in tables, for instance, data that is not yet seen as valid by business applications.

Potential uses of temporal tables include the following:

- Maintaining and accessing data applicable to different time periods.
- Associating a date and time range to row data.
- Enforcing date and time-based constraints. For instance, an employee can be assigned only to a single department in any given time frame.
- Updating or deleting a row for part of its validity period.
- Storing future data.

It is expensive and technically complex for businesses to develop their own temporal support infrastructure, such as additional tables, triggers, and application logic. By using temporal tables, businesses can store and retrieve their time-based data without having to build, maintain, and administer complex temporal infrastructure.

Related concepts:

"Time Travel Query using temporal tables" in *Database Administration Concepts and Configuration Reference*

RDF application development support has been added

DB2 Database for Linux, UNIX, and Windows now support the Resource Description Framework (RDF) to model information using Uniform Resource Identifiers (URIs). You can develop applications that easily store and query RDF data in DB2 databases.

RDF creates relationships between data in the form of triples or quads. Use DB2 RDF commands to create, modify, or delete your RDF stores. Use the SPARQL query language to query and modify data within these stores. You can also load RDF data in bulk into DB2 databases.

Related concepts:

"RDF application development for IBM data servers " in

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 10.1 features, you must upgrade to a Version 10.1 IBM data server client or driver.

In general, you can use Version 9.7 and Version 9.5 clients and drivers for running applications, developing applications, and performing database administration tasks on DB2 Version 10.1. Likewise, you can use Version 10.1 clients and drivers for running applications, developing applications, and performing administrative tasks on DB2 Version 9.8, Version 9.7, and Version 9.5 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 10.

DB2 V10 enhancements in IBM Data Server Driver for JDBC and SQLJ versions 3.62 and 4.12

The following IBM Data Server Driver for JDBC and SQLJ enhancements are available in version 3.62 or version 4.12, or later. Versions 3.62 and Version 4.12 were first shipped with DB2 Version 9.7 Fix Pack 4.

db2sqljprint support for metadata for temporal tables

The IBM Data Server Driver for JDBC and SQLJ db2sqljprint profile printer utility is enhanced to display metadata for temporal tables. For example:

```
...
Parameter 3:
  name:START_TS
  label:null
  nullable:false
  sqlType:392
  precision:0
  scale:0
  ccsid:37
  columnLength:32
  tableName:POLICY_DETAIL
  temporal column: ROW BEGIN
Parameter 4:
  name:END_TS
  label:null
  nullable:false
  sqlType:392
  precision:0
  scale:0
  ccsid:37
  columnLength:32
  tableName:POLICY_DETAIL
  temporal column: ROW END
Parameter 5:
  name:TRANS_ID
  label:null
```

```

nullable:true
sqlType:393
precision:0
scale:0
ccsid:37
columnLength:32
tableName:POLICY_DETAIL
temporal column: TRANSACTION START ID
...

```

DB2 V10 enhancements in IBM Data Server Driver for JDBC and SQLJ versions 3.63 and 4.13

The following IBM Data Server Driver for JDBC and SQLJ enhancements are available in version 3.63 or version 4.13, or later.

System monitoring enhancement

The server time that is returned by `DB2SystemMonitor.getServerTimeMicros` now includes commit and rollback time.

New method for changing an expired password

New method `com.ibm.db2.jcc.DB2Driver.changeDB2Password` can be used to change a password, whether or not the password is expired.

Better defaults for global configuration properties

Several defaults for global configuration properties were changed to values that are more appropriate for typical customer environments. The following table lists the old and new defaults.

Configuration property	Default before versions 3.63 and 4.13	Default for versions 3.63 and 4.13 or later
<code>db2.jcc.maxRefreshInterval</code>	30 seconds	10 seconds
<code>db2.jcc.maxTransportObjects</code>	-1 (unlimited)	1000
<code>db2.jcc.maxTransportObjectWaitTime</code>	-1 (unlimited)	1 second

Better defaults for Connection and DataSource properties

Several defaults for Connection and DataSource properties were changed to values that are more appropriate for typical customer environments. The following table lists the old and new defaults.

Connection and DataSource property	Default value before versions 3.63 and 4.13	Default value for versions 3.63 and 4.13 or later
<code>maxRetriesForClientReroute</code>	If <code>maxRetriesForClientReroute</code> and <code>retryIntervalForClientReroute</code> are not set, the connection is retried for 10 minutes, with a wait time between retries that increases as the length of time from the first retry increases.	If <code>maxRetriesForClientReroute</code> and <code>retryIntervalForClientReroute</code> are not set, <code>enableSysplexWLB</code> property is set to true, and the data server is DB2 for z/OS®, the default is 5. Otherwise, the default is the same as for previous driver versions.

Connection and DataSource property	Default value before versions 3.63 and 4.13	Default value for versions 3.63 and 4.13 or later
retryIntervalForClientReroute	If maxRetriesForClientReroute and retryIntervalForClientReroute are not set, the connection is retried for 10 minutes, with a wait time between retries that increases as the length of time from the first retry increases.	If maxRetriesForClientReroute and retryIntervalForClientReroute are not set, enableSysplexWLB property is set to true, and the data server is DB2 for z/OS, the default is 0 seconds. Otherwise, the default is the same as for previous driver versions.

Chapter 12. DB2 Text Search enhancements

Version 10.1 includes enhancements that extend the Text Search functionality.

The following enhancements are included:

- DB2 Text Search improvements (see “DB2 Text Search improvements for search”)
- DB2 Text Search supports stand-alone server deployment (see “DB2 Text Search improvements for search”)
- DB2 Text Search supports partitioned database environments (see “DB2 Text Search improvements for search”)

DB2 Text Search improvements for search

DB2 Text Search has been enhanced to support new search features and improved linguistic processing capabilities.

Search capabilities now include fuzzy and proximity search features. A fuzzy search is used to find words with spellings that are similar to that of the search terms. A proximity search retrieves documents containing search words which are located within a specified distance from each other.

DB2 Text Search now provides a morphological segmentation option (also called dictionary-based word segmentation) for Chinese, Japanese, and Korean languages. Morphological segmentation uses a language-specific dictionary to identify words in the sequence of characters in the document. This technique provides precise search results, because the dictionaries are used to identify word boundaries.

For wildcard searches like 'so*', a configuration parameter **queryExpansionLimit** is available to set the wildcard expansion limit. If the number of distinct matching terms in the text search index exceeds the limit, only the documents that match the already expanded terms are included in the search result.

Use a percentage sign (%) to specify that a term or phrase is optional instead of preceding a term with a '?'. For more information about DB2 Text Search arguments, see document.

Related concepts:

"Linguistic processing for DB2 Text Search" in Text Search Guide

"Fuzzy search" in Text Search Guide

"Proximity search " in Text Search Guide

Related reference:

"DB2 Text Search arguments" in Text Search Guide

DB2 Text Search stand-alone server deployment

DB2 Text Search now supports a stand-alone server setup in addition to an integrated setup.

A stand-alone text search server, also known as an Enterprise Content Management (ECM) Text Search server, can be installed and administered independent of the DB2 installation and may be located on the same or a different physical machine

than the database server. The stand-alone text search server is automatically enabled for support of rich text and proprietary formats.

A stand-alone DB2 Text Search server setup is preferred for partitioned environments as it facilitates workload distribution. Cross-platform deployment for the DB2 database server and text search server is also supported.

For both the integrated and the stand-alone server deployment, the text search server provides additional controls to tune resource usage, as well as finer control for logging and tracing.

Related concepts:

"DB2 Text Search server deployment scenarios" in Text Search Guide

DB2 Text Search supports partitioned database environments

DB2 Text Search supports full-text search in a partitioned database environment.

Each text search index will be partitioned into multiple text search index collections according to the partitioning of the table that hosts the index. A text search index update therefore results in multiple collection updates, one for each partition. It can be specified for each text search index whether the collection updates are executed in parallel or serially, the default is parallel. Capacity planning is required to determine resource needs.

Related concepts:

"DB2 Text Search in a partitioned database environment" in Text Search Guide

DB2 Text Search supports partitioned tables

DB2 Text Search supports full-text search on range-partitioned tables and tables that use the multidimensional clustering feature.

For non-partitioned database environments, the text search index for a partitioned table is mapped to a single text search index collection. In a partitioned database environment, it will be split into text search index collections according to the database partitioning.

Text search indexes on range-partitioned tables use integrity processing to identify changes. This same mechanism can be activated for non-partitioned tables by setting the **AUXLOG** index configuration to ON. Instead of using insert/delete triggers, the incremental update will use a text-maintained staging infrastructure for integrity processing, which includes for example, data from load insert.

This infrastructure will result in similar table status changes as the use of materialized query tables with deferred refresh. A new **db2ts RESET PENDING** text search command has been provided in this DB2 release as a convenience to run a set integrity statement on the affected dependent tables. If the text search index was not created by the user executing the command, and the user does not have DBADM privilege, the SECADM needs to assign CONTROL on the auxiliary log table.

Related concepts:

"Text search index creation, updates and property alterations" in Text Search Guide

"Incremental updates for DB2 Text Search indexes" in Text Search Guide

Related reference:

"db2ts RESET PENDING command" in Text Search Guide

Chapter 13. Installation and upgrade enhancements

Version 10.1 includes enhancements that make it faster to deploy products and easier to maintain them.

Product installations on all operating systems have been improved as follows:

- New command checks prerequisites (see “New db2prereqcheck command checks prerequisites before starting an installation” on page 83)
- New command parameters (see “Installation commands have been enhanced”)
- Installation of IBM DB2 pureScale Feature improved (see “DB2 pureScale Feature included in DB2 server edition installations” on page 84)
- IBM Data Studio can be installed from the DB2 Launchpad (see “Installation of IBM Data Studio integrated into the DB2 installation process” on page 84)
- New response file keywords (see “New response file keywords have been added” on page 83)
- License violations reported (see “DB2 License compliance report enhanced” on page 83)

If you have a Version 9.5, Version 9.7, or Version 9.8 copy already installed and you want to use Version 10.1 instead, you need to upgrade to Version 10.1. Version 10.1 is a new release. You cannot apply a fix pack to upgrade from a prior version to Version 10.1.

To learn about upgrade limitations, possible issues, and other details, see “Upgrade essentials for DB2 servers” in *Upgrading to DB2 Version 10.1* and “Upgrade essentials for clients” in *Upgrading to DB2 Version 10.1*.

Upgrading your DB2 servers and DB2 clients to Version 10.1 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 10.1* and “Upgrade essentials for routines” in *Upgrading to DB2 Version 10.1* topics.

Installation commands have been enhanced

Several installation-related commands have been enhanced to provide additional flexibility in installing and maintaining your DB2 environment.

New parameters have been added to the following installation-related commands:

Table 18. Summary of new installation-related command parameters

Installation command	New parameter
db2cluster_prepare	The following new parameters apply to a DB2 pureScale environment: -instance_shared_mount <i>Shared_Mounting_Dir</i> You can specify the mount point for a new General Parallel File System (GPFS™). -cfs_takeover With this parameter, you can indicate that the DB2 product take control of a cluster.

Table 18. Summary of new installation-related command parameters (continued)

Installation command	New parameter
<p>db2icrt db2iupdt</p>	<p>The new -j parameter makes configuring the DB2 Text Search server easier by using default values.</p> <ul style="list-style-type: none"> -j "TEXT_SEARCH" or -j "TEXT_SEARCH, servicename" or -j "TEXT_SEARCH, servicename, portnumber" or -j "TEXT_SEARCH, portnumber" <p>The following new parameters apply to a DB2 pureScale environment:</p> <p>-mnet <i>MemberNetName</i> Use this parameter to specify the cluster interconnect netname for the member. This new parameter replaces the <i>:netname</i> syntax of the -m <i>hostname:netname</i> parameter. The <i>:netname</i> syntax has been deprecated and might be removed in a future release. You specify the member host by using the -m <i>MemberHostName</i> parameter.</p> <p>-cfnet <i>CFNetName</i> Use this parameter to specify the cluster interconnect netname for the cluster caching facility (CF). This new parameter replaces the <i>:netname</i> syntax of the -cf <i>hostname:netname</i> parameter. The <i>:netname</i> syntax has been deprecated and might be removed in a future release. You specify the CF by using the -cf <i>CFHostName</i> parameter.</p> <p>-instance_shared_mount <i>Shared_Mounting_Dir</i> With this new parameter you can specify the mount point for a new General Parallel File System (GPFS).</p>
<p>db2iupgrade</p>	<p>Configuring the DB2 Text Search server is made easier using default values.</p> <ul style="list-style-type: none"> -j "TEXT_SEARCH" or -j "TEXT_SEARCH, servicename" or -j "TEXT_SEARCH, servicename, portnumber" or -j "TEXT_SEARCH, portnumber"
<p>db2nrupdt db2nrupgrade</p>	<p>Configuring the DB2 Text Search server is made easier using default values.</p> <ul style="list-style-type: none"> -j "TEXT_SEARCH" or -j "TEXT_SEARCH, portnumber"
<p>db2setup</p>	<p>New parameter -c validates the contents of a response file before you install a fix pack. The contents of a response file can now be validated without the need to perform the installation. This ensures the contents of the response file are correct and complete before deploying the installation.</p>
<p>db2val</p>	<p>New tracing parameter -t enables validation tracing and replaces the existing -d parameter. The -d parameter has been deprecated and might be removed in a future release.</p>
<p>installFixPack</p>	<p>The -p parameter installs a DB2 fix pack in a specified location. In addition, all license certificates are automatically applied. For example, if DB2 Workgroup Server Edition is installed in <i>path1</i> and you want to install the fix pack in <i>path2</i> and apply all of the license certificates in that path, issue the following command:</p> <pre>installFixPack -b <i>path1</i> -p <i>path2</i></pre> <p>In a DB2 pureScale environment, the new -H parameter applies a fix pack across multiple hosts. Certain restrictions apply. See the installFixPack command for details.</p>

Related reference:

"db2icrt - Create instance " in Command Reference

"db2iupdt - Update instances " in Command Reference

"db2setup - Install DB2 database products " in Command Reference

"db2_install - Install DB2 database product " in Command Reference

"installFixPack - Update installed DB2 database products " in Command Reference

"db2val - DB2 copy validation tool " in Command Reference

New response file keywords have been added

You can use new keywords during unattended installations that use response files.

Due to new features and functionality, Version 10.1 new keywords are available for unattended installations that use response files.

Table 19. Summary of new installation command parameters

Response file keyword	Details
INSTANCE_SHARED_MOUNT	Specifies the mount point for a new General Parallel File System (GPFS).
REMOVE_INSTALL_CREATED_USERS_GROUPS	Removes the users and groups that the DB2 installer creates for the current DB2 copy.
SSH_SERVER_INSTALL_DIR	On Windows operating systems, determines the base installation path for the IBM Secure Shell (SSH) Server for Windows service.
AUTOSTART_SSH_SERVER	On Windows operating systems, determines if the IBM Secure Shell (SSH) Server for Windows service should be autostarted

Related reference:

"Response file keywords" in Installing DB2 Servers

New db2prereqcheck command checks prerequisites before starting an installation

You can use the **db2prereqcheck** command to check whether your system meets the prerequisites for the installation of a specific DB2 version.

Using the **db2prereqcheck** command you can determine if your system satisfies the prerequisite requirements for the DB2 installation without having to download the new DB2 version and start the installation process.

DB2 License compliance report enhanced

The license compliance report now indicates which product and feature caused the violation.

Each DB2 product and feature has an associated license key. To verify the license compliance of your DB2 products and features, analyze a DB2 license compliance report. If there are any licensing violations, they can be addressed by obtaining the appropriate license keys or by removing the problematic DB2 products or features.

Related tasks:

"Analyzing DB2 license compliance reports" in Troubleshooting and Tuning Database Performance

DB2 pureScale Feature included in DB2 server edition installations

In DB2 Version 10, you can install the IBM DB2 pureScale Feature while installing DB2 Enterprise Server Edition, DB2 Workgroup Server Edition, and DB2 Advanced Enterprise Server Edition.

The DB2 pureScale Feature is supported only on AIX and Linux x86_64 operating systems.

You cannot install a DB2 product with the DB2 pureScale Feature in the same path as an existing DB2 Enterprise Server Edition, DB2 Workgroup Server Edition, or DB2 Advanced Enterprise Server Edition installation. Conversely, you cannot install DB2 Enterprise Server Edition, DB2 Workgroup Server Edition, or DB2 Advanced Enterprise Server Edition in the same path as an existing installation of a DB2 product with the DB2 pureScale Feature.

To install the DB2 pureScale Feature, use one of these methods:

- From the DB2 Launchpad, select the appropriate DB2 product with the DB2 pureScale Feature.
- Using a DB2 response file installation, specify a custom installation and select the DB2 pureScale Feature.

DB2 Spatial Extender is now included in the DB2 database product media

The DB2 Spatial Extender initial installation and fix pack application are now simplified by requiring one single installation media.

The DB2 Spatial Extender allows the storage and query of spatial data types of points, lines and polygons representing objects like highways, customer locations and parcel boundaries.

In Version 10.1, you can use the DB2 Setup wizard to install DB2 Spatial Extender as part of the DB2 database product installation. Spatial Extender is available when you select a custom installation. A separate CD or DVD for DB2 Spatial Extender is no longer available.

In earlier releases, the DB2 Spatial Extender had its own separate CD or DVD and it was not included in any DB2 database product media.

Related tasks:

"Setting up and installing DB2 Spatial Extender" in Spatial Extender User's Guide and Reference

Installation of IBM Data Studio integrated into the DB2 installation process

Following the installation of your DB2 product, you can now install IBM Data Studio from the DB2 Launchpad.

You can use the IBM Data Studio full client component for database administration tasks, analyzing and tuning queries, and creating, deploying, and debugging database applications.

You can use the IBM Data Studio web console component to monitor the health of your database, manage jobs, and share database catalog connection information between Data Studio clients. Authorized users can access the web console from a web browser or from within the Data Studio full client.

Related concepts:

"IBM Data Studio integrated installation with DB2 database products" in Installing DB2 Servers

Chapter 14. DB2 pureScale Feature enhancements

The IBM DB2 pureScale Feature was first introduced in Version 9.8. Version 10.1 builds on DB2 pureScale Feature support.

The following enhancements are included:

- Installation of DB2 pureScale Feature improved (see “DB2 pureScale Feature included in DB2 server edition installations” on page 84)
- Support added for AIX servers on RoCE networks (see “DB2 pureScale Feature support added for AIX servers on RoCE networks” on page 88)
- Support on Red Hat Enterprise Linux (RHEL) 6.1 has been added. For details, see “Network topology configuration” in *Installing DB2 Servers*.
- A fix pack can be installed across multiple hosts using the **installFixPack** command new **-p** parameter. See “Installation commands have been enhanced” on page 81.
- Range partitioning supported added (see “Range partitioning is now available for DB2 pureScale environments” on page 90)
- db2val command support added (see “DB2 pureScale Feature can be validated using db2val command” on page 90)
- **db2cluster** command now supports repairing an instance domain and controlling when an automatic failback occurs (see “**db2cluster** command now supports repairing an instance domain and controlling when an automatic failback occurs” on page 91)
- New CURRENT MEMBER default value improves DB2 pureScale performance (see “New CURRENT MEMBER default value improves DB2 pureScale performance” on page 92)
- New monitoring table function and elements improve global buffer pool performance (see “New monitoring routine provides insight into group buffer pool usage” on page 93)
- DB2 workload manager is now available (see “DB2 workload manager is now available in the DB2 pureScale environment” on page 69)
- Diagnostic data collection has been improved (see “Diagnostic data collection for DB2 pureScale environments has been improved” on page 93)

Note: In Version 10.1, occurrences of “DB2 pureCluster Feature” in documentation or messages refer to the IBM DB2 pureScale Feature.

DB2 pureScale Feature included in DB2 server edition installations

In DB2 Version 10, you can install the IBM DB2 pureScale Feature while installing DB2 Enterprise Server Edition, DB2 Workgroup Server Edition, and DB2 Advanced Enterprise Server Edition.

The DB2 pureScale Feature is supported only on AIX and Linux x86_64 operating systems.

You cannot install a DB2 product with the DB2 pureScale Feature in the same path as an existing DB2 Enterprise Server Edition, DB2 Workgroup Server Edition, or DB2 Advanced Enterprise Server Edition installation. Conversely, you cannot install DB2 Enterprise Server Edition, DB2 Workgroup Server Edition, or DB2

Advanced Enterprise Server Edition in the same path as an existing installation of a DB2 product with the DB2 pureScale Feature.

To install the DB2 pureScale Feature, use one of these methods:

- From the DB2 Launchpad, select the appropriate DB2 product with the DB2 pureScale Feature.
- Using a DB2 response file installation, specify a custom installation and select the DB2 pureScale Feature.

DB2 pureScale Feature support added for AIX servers on RoCE networks

Support for IBM DB2 pureScale Feature on AIX servers interconnected with a Remote Direct Memory Access (RDMA) over Converged Ethernet (RoCE) network has been added in DB2 for Linux, UNIX, and Windows Version 10.1.

Remote Direct Memory Access (RDMA) is required for cluster caching facilities and members to communicate over a high-speed, low latency cluster interconnect network. In previous releases of DB2 for Linux, UNIX, and Windows, support for DB2 pureScale Feature with an RDMA over Converged Ethernet (RoCE) cluster interconnect network was limited to Linux servers.

Support for RoCE cluster interconnect networks provides more choice for system architects deploying a DB2 pureScale Feature solution. By using existing Ethernet network infrastructure you can help reduce deployment costs, as you do not have to adopt a new network medium.

Installation commands have been enhanced

Several installation-related commands have been enhanced to provide additional flexibility in installing and maintaining your DB2 environment.

New parameters have been added to the following installation-related commands:

Table 20. Summary of new installation-related command parameters

Installation command	New parameter
db2cluster_prepare	The following new parameters apply to a DB2 pureScale environment: -instance_shared_mount <i>Shared_Mounting_Dir</i> You can specify the mount point for a new General Parallel File System (GPFS). -cfs_takeover With this parameter, you can indicate that the DB2 product take control of a cluster.

Table 20. Summary of new installation-related command parameters (continued)

Installation command	New parameter
<p>db2icrt db2iupdt</p>	<p>The new -j parameter makes configuring the DB2 Text Search server easier by using default values.</p> <ul style="list-style-type: none"> -j "TEXT_SEARCH" or -j "TEXT_SEARCH, servicename" or -j "TEXT_SEARCH, servicename, portnumber" or -j "TEXT_SEARCH, portnumber" <p>The following new parameters apply to a DB2 pureScale environment:</p> <p>-mnet <i>MemberNetName</i> Use this parameter to specify the cluster interconnect netname for the member. This new parameter replaces the <i>:netname</i> syntax of the -m <i>hostname:netname</i> parameter. The <i>:netname</i> syntax has been deprecated and might be removed in a future release. You specify the member host by using the -m <i>MemberHostName</i> parameter.</p> <p>-cfnet <i>CFNetName</i> Use this parameter to specify the cluster interconnect netname for the cluster caching facility (CF). This new parameter replaces the <i>:netname</i> syntax of the -cf <i>hostname:netname</i> parameter. The <i>:netname</i> syntax has been deprecated and might be removed in a future release. You specify the CF by using the -cf <i>CFHostName</i> parameter.</p> <p>-instance_shared_mount <i>Shared_Mounting_Dir</i> With this new parameter you can specify the mount point for a new General Parallel File System (GPFS).</p>
<p>db2iupgrade</p>	<p>Configuring the DB2 Text Search server is made easier using default values.</p> <ul style="list-style-type: none"> -j "TEXT_SEARCH" or -j "TEXT_SEARCH, servicename" or -j "TEXT_SEARCH, servicename, portnumber" or -j "TEXT_SEARCH, portnumber"
<p>db2nrupdt db2nrupgrade</p>	<p>Configuring the DB2 Text Search server is made easier using default values.</p> <ul style="list-style-type: none"> -j "TEXT_SEARCH" or -j "TEXT_SEARCH, portnumber"
<p>db2setup</p>	<p>New parameter -c validates the contents of a response file before you install a fix pack. The contents of a response file can now be validated without the need to perform the installation. This ensures the contents of the response file are correct and complete before deploying the installation.</p>
<p>db2val</p>	<p>New tracing parameter -t enables validation tracing and replaces the existing -d parameter. The -d parameter has been deprecated and might be removed in a future release.</p>
<p>installFixPack</p>	<p>The -p parameter installs a DB2 fix pack in a specified location. In addition, all license certificates are automatically applied. For example, if DB2 Workgroup Server Edition is installed in <i>path1</i> and you want to install the fix pack in <i>path2</i> and apply all of the license certificates in that path, issue the following command:</p> <pre>installFixPack -b <i>path1</i> -p <i>path2</i></pre> <p>In a DB2 pureScale environment, the new -H parameter applies a fix pack across multiple hosts. Certain restrictions apply. See the installFixPack command for details.</p>

Related reference:

"db2icrt - Create instance " in Command Reference

"db2iupdt - Update instances " in Command Reference

"db2setup - Install DB2 database products " in Command Reference

"db2_install - Install DB2 database product " in Command Reference

"installFixPack - Update installed DB2 database products " in Command Reference

"db2val - DB2 copy validation tool " in Command Reference

Range partitioning is now available for DB2 pureScale environments

You can now use range partitioning for DB2 pureScale tables.

With range partitioning, you can divide large table objects between multiple partitions for better performance.

You can use range partitioning in DB2 pureScale tables; this includes tables that use the PARTITION BY RANGE clause. In addition, the commands associated with range partitioning can be used in a DB2 pureScale environment.

This means, for example, that all of the following operations are supported:

- The roll-in and roll-out partition operations available through the ALTER TABLE statement
- The PARTITIONED and NOT PARTITIONED clauses for the CREATE INDEX statement
- For partitioned indexes, the ON DATA PARTITION clause of the REORG TABLE and REORG INDEXES ALL statements

In addition, the MON_GET_PAGE_ACCESS_INFO table function has been updated to work with range partitioned tables. All existing monitoring functions that operate against ranged partitioned tables will work for DB2 pureScale tables.

If you are already using the DB2 pureScale Feature, you can use range partitioning to help resolve page contention issues. By spreading contention out over a larger range, you can reduce data page contention; similarly, you can reduce contention with index pages by using partitioned indexes.

Related concepts:

"Table partitioning" in Partitioning and Clustering Guide

"Table partitioning in a DB2 pureCluster environment" in Partitioning and Clustering Guide

Related tasks:

"Adjusting database configuration parameters to meet DB2 pureCluster environment requirements" in Installing DB2 Servers

Related reference:

"MON_GET_PAGE_ACCESS_INFO table function - Get buffer pool page waiting information" in Administrative Routines and Views

DB2 pureScale Feature can be validated using db2val command

You can now use the **db2val** command to verify the core functionality of a DB2 pureScale environment. The command validates the installation and instances.

This command quickly ensures your DB2 pureScale environment is correctly configured by checking the state of installation files and instance setup.

Related reference:

"db2val - DB2 copy validation tool " in Command Reference

db2cluster command now supports repairing an instance domain and controlling when an automatic failback occurs

The **db2cluster** command can now be used to repair an instance domain. Certain failure situations can be recovered more quickly by re-creating the cluster manager domain and any cluster resources. In addition, you can use **db2cluster** to control when an automatic failback of a member to its home host occurs.

Repairing a domain

If a failure situation occurs with a DB2 pureScale instance which requires the cluster manager domain to be re-created, you can use the **db2cluster** command to re-create the domain and the resource models for all instances on the cluster. In this context, the command can only be run as the DB2 cluster services administrator.

The cluster manager domain is re-created using the same configuration (tiebreaker and host failure detection time) as the existing cluster manager domain.

To re-create the domain, run the following command:

```
db2cluster -cm -repair -domain domain-name
```

To use the **db2cluster** command to repair the domain of an instance, the domain must have been previously created by the **db2cluster** command in a Version 10.1 DB2 pureScale environment. Instance domains created by using the **db2haicu** command in Version 10.1 partitioned database environments or non-partitioned databases cannot be repaired with the **db2cluster** command.

Disabling automatic failback

In a DB2 pureScale environment, a reboot or failure of a host causes its member to be automatically moved to a guest host in restart light mode. When the host becomes available, automatic failback causes the relocated member to be immediately moved back onto its home host. As an example, administrators might want to control when automatic failback occurs to verify the health of the restarted home host before the member is moved back and reintegrated into the cluster. Without this control, the administrator is required to take the home host offline, thereby disrupting transactions for the short time it takes to move the member to a guest host and back.

Starting with DB2 Version 10.1, you can disable automatic failback by running the following command:

```
db2cluster -cm -set -option autofailback -value off
```

You can then use the **db2cluster** command to manually start the automatic failback of a member to its home host by running the following command:

```
db2cluster -cm -set -option autofailback -value on
```

If there is a host failure and the member is ready to automatically fail back to its home host, an alert is raised for that specific member if automatic failback has been disabled. The `db2instance -list` command can be used to show the presence of that alert. The `db2cluster -cm -list -alert` command can be used to provide information about that alert and how to start automatic failback.

Related tasks:

"Repairing the cluster manager domain" in Troubleshooting and Tuning Database Performance

"Disabling automatic member failback" in Data Recovery and High Availability Guide and Reference

Related reference:

"Troubleshooting options for the `db2cluster` command" in Troubleshooting and Tuning Database Performance

New CURRENT MEMBER default value improves DB2 pureScale performance

The `ALTER TABLE` and `CREATE TABLE` SQL statements now have a `CURRENT MEMBER` option in the allowed set of default values.

Default CURRENT MEMBER column

This change allows you to use the value of the `CURRENT MEMBER` special register as the default value for the column. This register value is retrieved when the `INSERT`, `UPDATE`, or `LOAD` action is run. You can then partition workloads based on that current member value, and therefore, reduce database contention in a DB2 pureScale environment.

One method to reduce contention is to add the `CURRENT MEMBER` column to the table with the `ALTER TABLE` statement, and then set the range partitioning on the table using that new column. With this approach, newly inserted rows stay local to a member. Consequently, a member has an affinity to specific rows in the table, and therefore the cost of synchronization across members is reduced.

If you experience index contention issues, you can add an implicitly hidden column that has a `CURRENT MEMBER` default and then use the information in that column to redefine your indexes.

In a DB2 pureScale environment, the static cost of system resource usage is directly proportional to the amount of active sharing that occurs between cluster members. Using the `CURRENT MEMBER` column to range partition a table or an index reduces that level of active sharing between members, and therefore delivers performance gains across the entire environment.

Related concepts:

"Hidden columns" in Database Administration Concepts and Configuration Reference

"Using CURRENT MEMBER default value in a DB2 pureCluster environment to improve contention issues" in Troubleshooting and Tuning Database Performance

Related reference:

"ALTER TABLE " in SQL Reference Volume 2

"CREATE TABLE " in SQL Reference Volume 2

New monitoring routine provides insight into group buffer pool usage

The new MON_GET_GROUP_BUFFERPOOL table function provides information that can help you determine whether the configured size for the group buffer pool (GBP) in a DB2 pureScale environment is properly sized.

If the GBP does not have sufficient space when attempting to register a page or write a page to the GBP, a GBP_FULL error occurs. You can use the MON_GET_GROUP_BUFFERPOOL table function to report the number of times that the GBP_FULL error occurs for a specific member, the currently connected member, or all members in the DB2 pureScale environment. If the value for GBP_FULL increases over a period of time, you may need to increase the size of the GBP.

Related reference:

"Formulas for calculating buffer pool hit ratios" in Troubleshooting and Tuning Database Performance

"MON_GET_GROUP_BUFFERPOOL " in Administrative Routines and Views

DB2 workload manager is now available in the DB2 pureScale environment

With DB2 Version 10.1, you are now able to use DB2 workload manager (DB2 WLM) to manage your workloads when the IBM DB2 pureScale Feature is enabled. A good workload management configuration aids in maximizing system efficiency and throughput, while helping you to achieve your business performance objectives.

Related concepts:

"Introduction to DB2 workload management concepts" in DB2 Workload Management Guide and Reference

Diagnostic data collection for DB2 pureScale environments has been improved

Additional diagnostic data is now collected for DB2 pureScale components by the enhanced **db2support** command. This improved diagnostic data collection can help speed the troubleshooting process in DB2 pureScale environments.

To help make the problem determination process faster in DB2 pureScale environments, the **db2support** command is improved to collect most diagnostic data that is specific to DB2 pureScale components, such as cluster manager, cluster file system and uDAPL, by default. The `-purecluster` or `-purescale` parameter of

the **db2support** command is also enhanced. If you now specify the `-purecluster` or `-purescale` option, additional diagnostic data that is specific to DB2 pureScale environments is collected.

The **db2support** command now also collects diagnostic data from all hosts, which include both members and cluster caching facility (CF), by default. This improvement ensures that all information that might be necessary during the troubleshooting process is collected by default.

To help you quickly search for any diagnostic data that is collected by the **db2support** command, the `db2support.html` file is now enhanced to include links of the data collected in the `db2support.html` file that point to its corresponding flat files in the subdirectory of the `db2support` package. A plain text version of the map file called `db2support.map` file is also included in the **db2support** package.

Related tasks:

"Collecting environment information with the `db2support` command" in *Troubleshooting and Tuning Database Performance*

Related reference:

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

Chapter 15. Multicultural support enhancements

DB2 V10.1 provides more options for working with multicultural data.

The following enhancements are included:

- Use new locale-sensitive UCA-based collations based on CLDR 1.8.1 (see “New locale-sensitive UCA-based collations based on CLDR 1.8.1”)
- Use new locales based on CLDR 1.8.1 (see “New locales based on CLDR 1.8.1” on page 96)
- Additional territory codes supported by DB2 client (see “Additional territory codes supported by IBM data server clients” on page 96)

New locale-sensitive UCA-based collations based on CLDR 1.8.1

Version 10.1 provides additional locale-sensitive UCA-based collations.

A locale-sensitive collation provides the expected ordering of data based on a specified locale, which includes information such as language and territory. These collations can also be tailored to provide not case-sensitive and accent-insensitive ordering.

The Unicode Collation Algorithm (UCA) provides a specification for comparing two Unicode strings in a way that conforms to the requirements of the Unicode Standard. Locale-sensitive collations in Version 10.1 are implemented by version 5.2 of the UCA. These collations are based on version 1.8.1 of the Common Locale Data Repository (CLDR).

You can also use locale-sensitive UCA-based collations with the `COLLATION_KEY_BIT` SQL scalar function.

New locales representing the following languages are added:

- Afrikaans
- Armenian
- Azerbaijani
- Bengali
- Bengali (Traditional)
- Hausa
- Igbo
- Kannada (Traditional)
- Konkani
- Sinhala
- Sinhala (Dictionary)
- Swahili
- Urdu
- Welsh
- Yoruba

Related reference:

"COLLATION_KEY_BIT " in SQL Reference Volume 1

New locales based on CLDR 1.8.1

Version 1.8.1 of the Common Locale Data Repository (CLDR) contains data for over 500 locales, based on 186 languages and 159 territories. Where locale support is provided in the DB2 product, you can use these locales in Version 10.1.

All locales represented in CLDR 1.8.1 can be used in the CURRENT LOCALE LC_TIME special register. They can also be used in the following locale-sensitive SQL scalar functions and XQuery functions.

- DAYNAME
- LOWER (locale sensitive)
- MONTHNAME
- NEXT_DAY
- ROUND
- ROUND_TIMESTAMP
- TIMESTAMP_FORMAT
- TRUNC_TIMESTAMP
- TRUNCATE or TRUNC
- UPPER (locale sensitive)
- VARCHAR_FORMAT
- lower-case XQuery function
- upper-case XQuery function

Additional territory codes supported by IBM data server clients

Users of IBM data server clients with defined territory codes see date, time, and decimal point formatting appropriate to their locale.

The following new territories are supported in Version 10.1:

- Armenia
- Georgia
- Kenya
- Nepal
- Sri Lanka
- Tanzania

Part 2. What's changed

What's changed includes information about changes in existing functionality from Version 9.7 and Version 9.8.

DB2 Version 10.1 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 Version 10.1.

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 a previous release might produce different results in Version 10.1.

Maintaining application compatibility across releases is a key priority. However, some behavior has to change to take advantage of new and changed functionality in the current release.

The following chapters describe the changed functionality, deprecated functionality, and discontinued functionality in Version 10.1 that can impact existing applications.

Chapter 16, “Administration changes summary,” on page 99

This chapter describes the changes in the existing DB2 functionality related to database administration.

Chapter 17, “Database setup and product installation changes summary,” on page 111

This chapter describes the changes in the existing DB2 functionality related to database setup and product installation.

Chapter 18, “Security changes summary,” on page 117

This chapter describes the changes in the existing DB2 functionality related to security.

Chapter 19, “Application development changes summary,” on page 119

This chapter describes the changes in the existing DB2 functionality related to application development.

Chapter 20, “DB2 command and SQL statement changes summary,” on page 131

This chapter describes the changes to DB2 CLP commands, DB2 system commands, and SQL statements to support new capabilities.

Chapter 21, “Deprecated functionality,” on page 139

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 22, “Discontinued functionality,” on page 155

This chapter lists features and functionality that are unsupported in Version 10.1.

Chapter 23, “Summary of deprecated and discontinued DB2 functionality in Version 10.1 and earlier releases,” on page 171

This chapter lists features and functionality that have been deprecated or discontinued in DB2 Version 10.1.

For information about changes of DB2 database products and features, see “Functionality in DB2 features and DB2 product editions”. The related licensing and marketing information is available in the DB2 for Linux, UNIX, and Windows home page at <http://www.ibm.com/software/data/db2/linux-unix-windows/>.

Related reference:

Appendix A, “Functionality in DB2 features and DB2 product editions,” on page 195

Chapter 16. Administration changes summary

Version 10.1 includes changed functionality that affects how you administer and work with DB2 databases.

Replication Center is now a stand-alone tool

The Replication Center is now a stand-alone tool. Installation default options and command to start the Replication Center have changed.

Details

In Version 10.1, the Replication Center is available as a stand-alone tool on Linux and Windows operating systems. In previous releases, it was grouped with other Administration tools such as the Control Center. The Administration tools have been discontinued.

In Version 10.1, the **db2rc** command is available to start the Replication Center. The **db2cc -rc** command that was available in previous releases has been discontinued.

On Windows operating systems, you can also click **Start > Programs > IBM DB2 > DB2 copy name > Replication Center**, where *DB2 copy name* indicates the name of the DB2 copy that you specified during installation.

In Version 10.1, the Replication Center is installed as part of the replication tools component by default in typical or custom installations. However, the compact installation no longer installs the replication tools component, which includes the Replication Center. In previous releases, the replication tools were a required component for compact installations for certain products.

All the Replication Center functionality of previous releases is still available and supported.

Resolution

To install the Replication Center, make sure that you select a typical or custom installation for any of the DB2 database products.

To start the Replication Center, issue the **db2rc** command. On Windows operating systems, you can also use the **Start** menu.

Collection of package list information has changed

The mechanisms that enable the collection of package list information by the unit of work event monitor have changed.

Details

You can enable the collection of package list information by using one of the following two mechanisms:

- Enable collection at the database level by setting the **mon_uow_data** and **mon_uow_pkglist** database configuration parameters. In Version 10.1, the **mon_uow_data** database configuration parameter has two values: NONE and BASE.

In previous releases, this parameter had three possible values: NONE, BASE, and PKGLIST. The **mon_uow_data** parameter is the parent parameter of the new **mon_uow_pkglist** and **mon_uow_exclist** database configuration parameters. You can use these two parameters, collectively, to collect package list information, executable ID information, or both.

- Enable collection for a specific workload by using the COLLECT UNIT OF WORK DATA clause of the CREATE WORKLOAD or ALTER WORKLOAD statement. The syntax of the clause has changed to specify the collection of package list information, executable ID information, or both. For details, see “ALTER WORKLOAD and CREATE WORKLOAD statements have changed”.

Resolution

If you have scripts or applications that set the **mon_uow_data** database configuration parameter to PKGLIST, set the **mon_uow_data** database configuration parameter to BASE and the **mon_uow_pkglist** database configuration parameter to ON as shown in the following example:

```
UPDATE DB CFG FOR SAMPLE USING mon_uow_data BASE
UPDATE DB CFG FOR SAMPLE USING mon_uow_pkglist ON
```

If you upgrade a database that has the **mon_uow_data** database configuration parameter set to PKGLIST, the **mon_uow_data** database configuration parameter is set to BASE and the **mon_uow_pkglist** database configuration parameter is set to ON during the database upgrade.

If you specify the COLLECT UNIT OF WORK DATA clause for the CREATE WORKLOAD or ALTER WORKLOAD statement, start using the new syntax for this clause.

DB2 Text Search security model changes

DB2 Text Search now executes administrative operations based on the authorization ID of the user executing the operation.

Details

The instance owner does not require any prerequisites for database privileges anymore, and it is not necessary for the fenced user to be in the same primary group as the instance owner. Executing operations with the authorization ID of the user improves auditability and improves control of text search management.

To simplify access control, three new system roles are available:

- Text Search Administrator (SYSTS_ADM) - executes operations on database level
- Text Search Manager (SYSTS_MGR) - executes operations on index level
- Text Search User (SYSTS_USR) - has access to text search catalog data

Resolution

Change your scripts and applications to use the new DB2 Text Search security model.

DB2 Text Search index location has changed

The default location for text search indexes has changed.

Details

The default location for text search indexes is now controlled through the **defaultDataDirectory** parameter and can be set with the configTool utility. The parameter has no default configuration. If unchanged, collections will be created in a subfolder of the sql1ib folder, and not in the database path.

Resolution

Use the configTool utility to configure the new **defaultDataDirectory** parameter or change your scripts and applications to use custom collection directories.

See details in the description for **CREATE INDEX FOR TEXT** command.

DB2 Text Search scheduler changes

DB2 Text Search now uses the administrative task scheduler to automate the execution of tasks.

Details

Scheduling tasks are automatically created and updated with the **UPDATE FREQUENCY** settings for the text search index. You can monitor the task list and the status of executed tasks by using administrative views for the scheduler. For more information about the scheduler, see the topic about using the administrative task scheduler.

The schedule for a text index is only visible to the user who created the schedule and to users with DBADM privileges. If a text index that has a schedule is dropped by a user who is not the creator of the text index, nor has DBADM privileges, the drop will complete successfully, but the schedule task remains. Similarly, if such a user alters a text index to remove an existing schedule, the schedule task will remain. This arises if multiple users cross-manage text indexes and do not have DBADM privileges, since the authorization ID of the user is used to execute the administrative operations. So, to remove these orphaned schedules, connect with DBADM privilege, check the scheduler task list, and remove any orphaned schedule tasks.

Resolution

For text search indexes created with earlier releases, verify scheduled tasks and use the **ALTER INDEX** operation to set and update schedules.

DB2 Text Search administrative commands and stored procedures have changed

DB2 Text Search administrative commands and stored procedures have been added or modified to support new features in Version 10.1.

Details

Enhanced text search command have been provided in this DB2 release to support additional features. The following is a list of enhanced text search commands:

- db2ts ALTER INDEX FOR TEXT command
- db2ts CLEANUP FOR TEXT command

- db2ts CLEAR COMMAND LOCKS FOR TEXT command
- db2ts CLEAR EVENTS FOR TEXT command
- db2ts CREATE INDEX FOR TEXT command
- db2ts DISABLE DATABASE FOR TEXT command
- db2ts DROP INDEX FOR TEXT command
- db2ts ENABLE DATABASE FOR TEXT command
- db2ts RESET PENDING FOR TEXT command
- db2ts START FOR TEXT command
- db2ts STOP FOR TEXT command
- db2ts UPDATE INDEX FOR TEXT command

The following is a list of enhanced text search stored procedures:

- SYSTS_ADMIN_CMD procedure
- SYSTS_ALTER procedure
- SYSTS_CLEAR_EVENTS procedure
- SYSTS_CLEAR_COMMANDLOCKS procedure
- SYSTS_CONFIGURE procedure
- SYSTS_CREATE procedure
- SYSTS_DISABLE procedure
- SYSTS_DROP procedure
- SYSTS_ENABLE procedure
- SYSTS_UPDATE procedure

Resolution

Review the list of changed DB2 Text Search features and functionality to determine if your applications or scripts are impacted and then modify them accordingly.

Take advantage of changed functionality or features through the use of new text search parameters or new values for existing text search parameters.

Names of TOTALDBPARTITIONCONNECTIONS and TOTALSCPARTITIONCONNECTIONS thresholds have been changed

The name of the TOTALDBPARTITIONCONNECTIONS threshold has been changed to the TOTALMEMBERCONNECTIONS threshold. The name of the TOTALSCPARTITIONCONNECTIONS threshold has been changed to the TOTALSCMEMBERCONNECTIONS threshold.

Details

The functionality of each of the renamed thresholds remains the same. Only the names of the thresholds have changed. In partitioned database environments, database partitions are now referred to as members.

Resolution

Use the TOTALMEMBERCONNECTIONS threshold instead of the TOTALDBPARTITIONCONNECTIONS threshold.

Use the TOTALSCMEMBERCONNECTIONS threshold instead of the TOTALSCPARTITIONCONNECTIONS threshold.

Section actuals now include object statistics

In Version 10.1, section actuals information in the explain facility output includes statistics for database objects.

Details

In previous releases, output from the explain facility included actuals information only for operators. This output now includes runtime statistics for tables and indexes that a statement section accesses when it executes.

Resolution

Before you can obtain a section explain with actuals that includes object statistics, you must migrate your explain tables. To migrate explain tables, use the SYSINSTALLOBJECTS procedure or **db2exmig** command.

The default diagnostic log paths are now split by default

In Version 10.1, all members, cluster caching facilities, database partition servers, and database partitions log to a private db2diag.log by default.

Details

In previous versions, all members and cluster caching facilities logged their diagnostic data to a common location on a GPFS share under the directory sqllib_shared. To have members, cluster caching facilities, database partition servers, and database partitions log to separate diagnostic directories, you had to manually specify that diagnostic data was to be split.

This change to private diagnostic directories improves the diagnostic logging performance because it results in less contention on the db2diag.log and cfdiag.*.log files. As well, it prevents a single point of failure.

Table 21 and Table 22 on page 104 show the diagnostic paths for newly created Version 10.1 instances that are updated using the following command:

```
update dbm cfg using [DIAGPATH|CF_DIAGPATH|ALT_DIAGPATH] <config_setting>
```

Note that the actual values that are stored and displayed in the NULL input case are the actual default value, effectively meaning that the NULL setting is no longer support. That's to say, even if you specify the NULL setting, the configuration file is populated with the default value.

Table 21. Setting for the diag.log and cfdiag.*.log paths in a new DB2 instance

config_setting	Single partitioned environment	Partitioned database environment	DB2 pureScale environment
NULL	<code>\$INSTHOME/sqllib/db2dump/ \$m</code>	<code>\$INSTHOME/sqllib/db2dump/ \$m</code>	<code>\$INSTHOME/sqllib_shared/db2dump/ \$m</code>
<i>path</i>	<i>path</i>	<i>path</i>	<i>path</i>

Table 21. Setting for the *diag.log* and *cfdiag.*.log* paths in a new DB2 instance (continued)

config_setting	Single partitioned environment	Partitioned database environment	DB2 pureScale environment
\$X ¹	\$INSTHOME/sqllib/db2dump/ \$X	\$INSTHOME/sqllib/db2dump/ \$X	\$INSTHOME/sqllib_shared/db2dump/ \$X
<i>path</i> \$X	<i>path</i> \$X	<i>path</i> \$X	<i>path</i> \$X
\$X/ <i>path</i>	\$INSTHOME/sqllib/db2dump/ \$X/ <i>path</i>	\$INSTHOME/sqllib/db2dump/ \$X/ <i>path</i>	\$INSTHOME/sqllib_shared/db2dump/ \$X/ <i>path</i>
<i>path</i> \$X/ <i>path</i>	<i>path</i> \$X/ <i>path</i>	<i>path</i> \$X/ <i>path</i>	<i>path</i> \$X/ <i>path</i>
¹ \$X represents one of the following split diagpath tokens: <ul style="list-style-type: none"> • \$h • \$n • \$m • \$h\$n • \$h\$m 			

Because the shared directory path and \$INSTHOME are now separated, there is a default **alt_diagpath** that is physically separated from the **diagpath**, which was the previously recommended configuration.

Table 22. Setting for the alternative *diag* path in a new DB2 instance

config_setting	Single partitioned environment	Partitioned database environment	DB2 pureScale environment
NULL	" "	" "	\$INSTHOME/sqllib/db2adump/ \$m
<i>path</i>	<i>path</i>	<i>path</i>	<i>path</i>
\$X	\$INSTHOME/sqllib/db2dump/ \$X	\$INSTHOME/sqllib/db2dump/ \$X	\$INSTHOME/sqllib_shared/db2adump/ \$X
<i>path</i> \$X	<i>path</i> \$X	<i>path</i> \$X	<i>path</i> \$X
\$X/ <i>path</i>	\$INSTHOME/sqllib/db2dump/ \$X/ <i>path</i>	\$INSTHOME/sqllib/db2dump/ \$X/ <i>path</i>	\$INSTHOME/sqllib/db2adump/ \$X/ <i>path</i>
<i>path</i> \$X/ <i>path</i>	<i>path</i> \$X/ <i>path</i>	<i>path</i> \$X/ <i>path</i>	<i>path</i> \$X/ <i>path</i>

Table 23 shows what the settings would be for the **diagpath** and **cf_diagpath** configuration parameters following an upgrade from a pre-Version 10.1 DB2 instance.

Table 23. Setting for the *diag.log* and *cfdiag.*.log* paths in an upgraded DB2 instance

config_setting	Single partitioned environment	Partitioned database environment	DB2 pureScale environment
NULL	\$INSTHOME/sqllib/db2dump	\$INSTHOME/sqllib/db2dump	\$INSTHOME/sqllib_shared/db2dump
<i>path</i>	<i>path</i>	<i>path</i>	<i>path</i>
\$X ¹	\$INSTHOME/sqllib/db2dump/ \$X	\$INSTHOME/sqllib/db2dump/ \$X	\$INSTHOME/sqllib_shared/db2dump/ \$X

Table 23. Setting for the diag.log and cfdiag.*.log paths in an upgraded DB2 instance (continued)

config_setting	Single partitioned environment	Partitioned database environment	DB2 pureScale environment
path \$X	path \$X	path \$X	path \$X
\$X/path	\$INSTHOME/sqllib/db2dump/ \$X/path	\$INSTHOME/sqllib/db2dump/ \$X/path	\$INSTHOME/sqllib_shared/db2dump/ \$X/path
path \$X/path	path \$X/path	path \$X/path	path \$X/path

Resolution

No user action is required, unless you want to specify a single diagpath directory. If you want to revert to the previous behavior of having a single location for diagnostic data, specify a diagnostic path without the token

Some database manager configuration parameters have been changed

Version 10.1 contains a number of new and changed database manager configuration parameters.

Details

New database manager configuration parameters

Due to new features and functionality, Version 10.1 contains a number of new configuration parameters.

Table 24. Summary of new Version 10.1 database manager configuration parameters

Parameter name	Description	Details
comm_exit_list	Communication buffer exit library list	This parameter specifies the list of communication buffer exit libraries that DB2 will use. A communication buffer exit library is a dynamically loaded library that vendor applications can use to gain access to and examine DB2 communication buffers used to communicate with client applications.
wlm_dispatcher	Workload manager dispatcher	This parameter enables (YES) or disables (NO) the DB2 workload manager (WLM) dispatcher. By default, an enabled WLM dispatcher controls only CPU limits.
wlm_disp_concur	Workload manager dispatcher thread concurrency	This parameter specifies how the DB2 workload manager (WLM) dispatcher sets the thread concurrency level. You can also manually set the thread concurrency level to a fixed value.
wlm_disp_cpu_shares	Workload manager dispatcher CPU shares	This parameter enables (YES) or disables (NO) the control of CPU shares by the DB2 workload manager (WLM) dispatcher. By default, an enabled WLM dispatcher controls only CPU limits.
wlm_disp_min_util	Workload manager dispatcher minimum CPU utilization	This parameter specifies the minimum amount of CPU utilization that is necessary for a service class to be included in the DB2 WLM-managed sharing of CPU resources.

Changed database manager configuration parameters

The following table lists the database manager configuration parameters with changes to their default values.

Table 25. Summary of configuration parameters with changed default values

Parameter name	Description	Details about default value change
alt_diagpath	Alternate diagnostic data directory path configuration parameter	<p>Previous releases Null</p> <p>Version 10.1 DB2 pureScale environments (Linux and UNIX) <i>INSTHOME</i>/sql1lib/db2adump/ \$m</p>
cf_diagpath	Diagnostic data directory path configuration parameter for the CF	<p>Previous releases of DB2 pureScale environments Null</p> <p>Version 10.1 DB2 pureScale environments <i>INSTHOME</i>/sql1lib/db2dump/ \$m</p>
diagpath	Diagnostic data directory path configuration parameter	<p>The default value of diagnostic data directory path configuration parameter is changed</p> <p>Previous releases Null</p> <p>Version 10.1 Server Edition (Linux and UNIX) <i>INSTHOME</i>/sql1lib/db2dump/</p> <p>Version 10.1 partitioned database environments (Linux and UNIX) <i>INSTHOME</i>/sql1lib/db2dump/ \$m</p> <p>Version 10.1 DB2 pureScale environments (Linux and UNIX) <i>INSTHOME</i>/sql1lib/db2dump/ \$m</p> <p>This new default value means that all database partitions, CFs, and members have their own diagnostic log directory.</p> <p>Version 10.1 (Windows) On partitioned database environments: Instance subdirectory of the directory specified by the <i>DB2INSTPROF</i> registry variable\ \$m</p> <p>Outside of partitioned database environments: Instance subdirectory of the directory specified by the <i>DB2INSTPROF</i> registry variable\</p> <p>You can use the new value \$m, which resolves to <i>DIAG_number</i>, to specify a unique diagnostic log path for all database partitions, CFs, or members.</p>
mon_obj_metrics	Monitoring object metrics configuration parameter	<p>Previous releases BASE</p> <p>Version 10.1 EXTENDED</p>

The following database manager configuration parameters have changed behaviors or have new ranges in Version 10.1.

Table 26. Summary of database manager configuration parameters with changed behaviors, new ranges, or new values

Parameter name	Description	Version 10.1 change
diagpath	Diagnostic data directory path configuration parameter	<ul style="list-style-type: none"> You can use the new value \$m, which resolves to DIAG_number, to specify a unique diagnostic log path for all database partitions, CFs, or members. The value \$n is deprecated and might be removed in a future release. Use the \$m value instead.
mon_obj_metrics	Monitoring object metrics configuration parameter	<ul style="list-style-type: none"> A new value, EXTENDED, has been added for this configuration parameter. You can now choose between three levels of data collection for object metrics: none, BASE and EXTENDED; the default value is EXTENDED.
numdb	Maximum number of concurrently active databases including host and System i [®] databases configuration parameter	<ul style="list-style-type: none"> The maximum number of concurrently active databases in a DB2 pureScale environment is now 200.

Resolution

Take advantage of enhanced functionality or new features by adopting new functionality through the use of new database manager configuration parameters or new values for existing database manager configuration parameters.

For new database manager configuration parameters or changes to existing database manager configuration parameters that result in DB2 server behavior changes, adjust your existing applications or scripts.

Some registry and environment variables have changed

There are a number of changes to registry and environment variables to take advantage of new and changed functionality in Version 10.1.

Details

New variables

These environment and registry variables are new in Version 10.1:

Table 27. Environment and Registry variables added

Registry variable	Description
DB2_INDEX_PCTFREE_DEFAULT	You can use this registry variable to specify what percentage of each index page to leave as free space when building the index.

Table 27. Environment and Registry variables added (continued)

Registry variable	Description
DB2_MCR_RECOVERY_PARALLELISM_CAP	You can use this registry variable to limit how many active databases are recovered in parallel on a member during member crash recovery. You can use this variable only in a DB2 pureScale environment.
DB2_XSLT_ALLOWED_PATH	You can use this registry variable to controls whether the DB2 instance refers to external entities defined inside of an XSLT stylesheet.

New values in existing registry variables

The following table shows the registry variables that have new values for Version 10.1 instances:

Table 28. Registry variables with new values

Registry variable	New values
DB2_PMODEL_SETTINGS	You can now use this variable's new SRVLST_EQUAL_WEIGHT option to override the default behavior in which member weights are computed based on load, and have non-zero member weights in the server list always be identical.

Changed behaviors

The following table describe the changes to registry variables when you create or upgrade to a Version 10.1 instance:

Table 29. Registry variables with changed behaviors

Registry variable	Changed behavior
DB2BPVARS	The NUMPREFETCHQUEUES and the PREFETCHQUEUESIZE option of this variable have been discontinued as optimization improvements render these options obsolete. The DB2BPVARS registry variable is still deprecated.
DB2_EXTENDED_OPTIMIZATION	The ENHANCED_MULTIPLE_DISTINCT setting has been deprecated and might be removed in a future release. Version 10.1 includes enhancements to improve the performance of multiple distinct queries. These enhancements are supported in all environments including partitioned database and DB2 pureScale environments. These enhancements are enabled by default in newly created databases and upgraded databases where this registry variable is not set to ENHANCED_MULTIPLE_DISTINCT . If you keep this setting, the previous release behavior is maintained. You should remove this setting to make the new enhancements available and benefit from better query performance.
DB2_NO_FORK_CHECK	This registry variable is no longer deprecated. Continue to use this variable to have the DB2 runtime client minimize checks to determine if the current process is a result of a fork call.
DB2NTNOCACHE	This registry variable is no longer deprecated. Continue to use this variable to override the undocumented 192 MB limit for the cache.

Resolution

Take advantage of enhanced functionality or new features by adopting new functionality through the use of new registry variables or new values for existing registry variables.

For new registry variables or changes to existing registry variables that result in DB2 server behavior changes, adjust your existing applications or scripts.

Review the list of deprecated and discontinued registry variables to learn about additional changes that might impact your applications and scripts and determine whether you have to adjust them or not.

Chapter 17. Database setup and product installation changes summary

Version 10.1 includes changed functionality that affects how you install DB2 database products and setup DB2 databases.

To take advantage of the new features included in the current release, some of the minimum software requirements have been updated. To ensure that your systems are correctly setup, review “Installation requirements for DB2 database products” and “Support for elements of the database application development environment”.

You can upgrade DB2 server or client copies to Version 10.1 from Version 9.8, Version 9.7, or Version 9.5. Version 10.1 is a new release and you cannot apply a fix pack to upgrade from previous releases.

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 10.1*.

Upgrading your DB2 servers and clients to Version 10.1 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 10.1* to help you determine whether there is any upgrade impact.

Informational constraints can now be TRUSTED or NOT TRUSTED

You can now specify the TRUSTED or NOT TRUSTED keywords when creating informational constraints for referential integrity constraints in tables and nicknames.

Details

In Version 10.1, informational constraints can be defined as TRUSTED or NOT TRUSTED. A NOT ENFORCED TRUSTED constraint has the same behavior as previous releases and it is the default. A NOT ENFORCED NOT TRUSTED constraint indicates that the data cannot be trusted to conform to the constraint. If a NOT ENFORCED NOT TRUSTED constraint is enabled for query optimization, then it will not be used to perform optimizations that depend on the data conforming completely to the constraint.

You can indicate the TRUSTED or NOT TRUSTED keywords after the NOT ENFORCED parameter in the following SQL statements only for referential integrity constraints:

- CREATE TABLE
- ALTER TABLE
- CREATE NICKNAME
- ALTER NICKNAME

Resolution

You should modify your applications and scripts and replace NOT ENFORCED with NOT ENFORCED TRUSTED. Although NOT ENFORCED and NOT ENFORCED TRUSTED are equivalents, it is a good practice to fully qualify SQL statements.

Modify your applications and scripts and add the NOT TRUSTED keyword for cases where the data conforms to the constraint for most rows, but it is not independently known that all the rows or future additions will conform to the constraint.

IBM Tivoli System Automation for Multiplatforms (SA MP) is now automatically installed

IBM Tivoli® System Automation for Multiplatforms (SA MP) is now automatically installed when SA MP is a required component for other components selected for installation.

Details

If you select any component that requires SA MP for installation, SA MP is automatically installed whether SA MP is selected or not. For example, IBM DB2 pureScale Feature requires SA MP. If you install DB2 pureScale Feature, SA MP is automatically installed.

As a result, the INSTALL_TSAMP response file keyword for DB2 installation is deprecated. If you indicate INSTALL_TSAMP=NO in the response file, SA MP is not installed only if any other selected components for installation do not require SA MP.

Resolution

Remove the INSTALL_TSAMP keyword from response files. SA MP is only installed when required by other components selected for installation.

Some database configuration parameters have been changed

Version 10.1 contains a number of new and changed database configuration parameters.

New database configuration parameters

Due to new features and functionality, Version 10.1 contains a number of new database configuration parameters.

Table 30. New Version 10.1 database configuration parameters

Parameter name	Description	Details
dft_schemas_dcc	Default data capture on new schemas	This parameter allows the control of default setting for DATA CAPTURE CHANGES on newly created schemas for replication purposes. For more information, see "Replication improvements" on page 50.

Table 30. New Version 10.1 database configuration parameters (continued)

Parameter name	Description	Details
hadr_replay_delay	HADR standby log replay time delay	This parameter specifies the time that must have passed from when the data is changed on primary before these changes would be reflected on the standby database. The time is specified in number of seconds. For more information, see “HADR delayed replay provides protection against application errors” on page 49.
hadr_spool_limit	HADR spool limit	This parameter allows log replay on the HADR standby database to be behind the HADR primary database. If there is a spike in transaction volume or slow replay caused by specific operations and the log receive buffer fills up, the log data is written (or <i>spooled</i>) to disk and then read later. For more information, see “HADR log spooling prevents spikes on throughput” on page 49.
hadr_target_list	HADR target list	This parameter, which is used to enable multiple HADR standbys, specifies a list of up to three target host:port pairs that act as HADR standby databases. For more information, see “HADR now supports multiple standby databases” on page 47.
log_appl_info	Application information log record	This parameter specifies that the application information log record is written at the start of each update transaction.
log_ddl_stmts	Log DDL statements	This parameter specifies that extra information regarding DDL statements will be written to the log.
mon_uow_execlist	Monitoring unit of work events with executable list	This parameter enables (ON) or disables (OFF) the collection of execution list information by the unit of work event monitor. By default, execution list information is not collected (OFF). It is a child parameter of the mon_uow_data database configuration parameter. For more information, see “Executable ID list is included in unit of work event monitor information” on page 31.
mon_uow_pkglist	Monitoring unit of work events with package list	This parameter enables (ON) or disables (OFF) the collection of package list information by the unit of work event monitor. By default, package list information is not collected (OFF). It is a child parameter of the mon_uow_data database configuration parameter. For more information, see “Collection of package list information has changed” on page 99.

Table 30. New Version 10.1 database configuration parameters (continued)

Parameter name	Description	Details
<code>system_time_period_adj</code>	Adjust temporal SYSTEM_TIME period	This database configuration parameter specifies how to handle the situation of a history row for a system-period temporal table potentially being generated with an end timestamp less than the begin timestamp.

Changed database configuration parameters

The following database configuration parameters have changed behaviors, new ranges, or new values in Version 10.1.

Table 31. Database configuration parameters with changed behaviors, new ranges, or new values

Parameter name	Description	Details
<code>auto_stats_views</code>	Statistical views configuration parameter	This parameter enables and disables automatic statistic collection on statistical views. When enabled, DB2 will maintain the statistics on statistical views automatically.
<ul style="list-style-type: none"> • <code>hadr_local_host</code> • <code>hadr_local_svc</code> • <code>hadr_peer_window</code> • <code>hadr_remote_host</code> • <code>hadr_remote_inst</code> • <code>hadr_remote_svc</code> • <code>hadr_syncmode</code> • <code>hadr_timeout</code> 	HADR-related configuration parameters	<p>In previous releases, no HADR configuration parameter could be updated dynamically; the database had to be deactivated and reactivated for the updates to take effect. Starting in Version 10.1, updates to these configuration parameters can take effect on the HADR primary without deactivating the database. You do have to issue a STOP HADR on the primary, followed by a START HADR AS PRIMARY. As a result, you can make configuration parameter updates to your HADR primary without having an impact on the applications that are using the database.</p> <p>Note: The following new HADR configuration parameters also have this behavior:</p> <ul style="list-style-type: none"> • <code>hadr_replay_delay</code> • <code>hadr_spool_limit</code> • <code>hadr_target_list</code>

Table 31. Database configuration parameters with changed behaviors, new ranges, or new values (continued)

Parameter name	Description	Details
mon_uow_data	Monitoring units of work	In Version 10.1, the values that you can specify for mon_uow_data have changed. The default value for mon_uow_data continues to be NONE. It is a parent parameter to mon_uow_exclist and mon_uow_pkglist . For more information, see "Collection of package list information has changed" on page 99.
num_iocleaners	Controls the number of asynchronous page cleaners for a database	In previous releases, the AUTOMATIC setting calculated the number of page cleaners based on the number of logical CPUs. In Version 10.1, the calculation now uses the number of physical CPU cores instead. (On the HP-UX platform the number of logical CPUs is still used.)
seqdetect	Sequential detection and readahead flag	This parameter controls whether the database manager is allowed to perform sequential detection or readahead prefetching during I/O activity.
mon_req_metrics	Monitoring request metrics configuration parameter	In Version 10.1, the default value for mon_req_metrics is changed from BASE to NONE.
mon_act_metrics	Monitoring activity metrics configuration parameter	The default value for mon_act_metrics is changed from BASE to NONE.
mon_obj_metrics	Monitoring object metrics configuration parameter	The default value for mon_obj_metrics is changed from BASE to NONE.
mon_lw_thresh	Monitoring lock wait threshold configuration parameter	The default value for mon_lw_thresh is changed from 5000000 to 4294967295.

Resolution

Take advantage of enhanced functionality or new features by adopting new functionality through the use of new database configuration parameters or new values for existing database configuration parameters.

For new database configuration parameters or changes to existing database configuration parameters that result in DB2 server behavior changes, adjust your existing applications or scripts.

Review the list of discontinued database configuration parameters to determine additional changes that might impact your applications and scripts.

Chapter 18. Security changes summary

Version 10.1 includes changed functionality that affects the scope and abilities of SYSADM, SECADM and DBADM authority levels, SSL configuration, and other features.

Kerberos authentication changes (UNIX)

Kerberos authentication support for several UNIX operating systems has changed in this release.

Details

On the Solaris and Linux operating systems, Kerberos authentication for DB2 connections now uses the MIT "krb5" Kerberos packages provided by the underlying operating system. These MIT Kerberos packages replace the ones provided by the IBM NAS Toolkit.

On the HP operating system, Kerberos authentication for DB2 connections is now supported using the MIT "krb5" Kerberos packages provided by the underlying operating system.

On the AIX operating system, Kerberos authentication for DB2 connections continues to use the Kerberos packages provided by the IBM NAS Toolkit.

The installation and configuration instructions for Kerberos have been updated accordingly. See [Installing and configuring Kerberos](#)

Resolution

Users on Solaris and Linux operating systems will need to first remove the Kerberos packages provided by the IBM NAS Toolkit, then install the MIT "krb5" Kerberos packages provided by the operating system.

For additional steps on installing and configuring the new Kerberos packages, see the installation instructions for your specific operating system in the [Related Links](#) section .

Chapter 19. Application development changes summary

Version 10.1 includes changed functionality that affects how you develop applications.

Also, review deprecated and discontinued functionality to determine whether it affects your existing applications.

XML casting error (SQL16061N) less likely to occur

To avoid unnecessary interruptions while querying XML data, casting and truncation is now handled in a similar way as for SQL.

Details

In versions prior to DB2 V10.1, simple errors in casting and comparison of XML data resulted in an SQLCODE being raised. These errors could be caused by specifying a data type that is too small, or comparing incompatible types of data. For example, in previous releases, both the following queries return error SQL16061N:

- `XMLCAST(XMLQUERY('Jonathan') AS VARCHAR(2))`
This query fails because the string, Jonathan, is too large to fit into a VARCHAR(2) value.
- `Xquery let $doc := <a>N/A return $doc[b < 3.4];`
This comparison fails because the query compares the character string 'N/A' with the number, 3.4.

Resolution

In DB2 V10.1, when XML data is cast to a CHAR, VARCHAR, or DECIMAL type, there are cases where message SQL16061N will no longer be issued for simple casting and comparison mistakes. The following table compares behavior in prior releases with the new behavior provided by DB2 V10.1.

Table 32. Comparison of pre-Version 10.1 behavior with DB2 V10.1 behavior

Query fragment	Releases prior to DB2 V10.1	DB2 V10.1
<code>XMLCAST(XMLQUERY('Jonathan') AS VARCHAR(2))</code>	Returns error SQL16061N	Query returns: Jo Warning SQL0445W is issued, because non-blank characters are truncated
<code>XMLCAST(XMLQUERY('Jo ') AS VARCHAR(2))</code>	Returns error SQL16061N	Query returns: Jo No warning is issued, because only blank characters are truncated
<code>XMLCAST (XMLQUERY('1.234') AS DECIMAL(3,2))</code>	Returns error SQL16061N	Query returns: 1.23

Table 32. Comparison of pre-Version 10.1 behavior with DB2 V10.1 behavior (continued)

Query fragment	Releases prior to DB2 V10.1	DB2 V10.1
Xquery let \$doc := <a>N/A return \$doc[b < 3.4];	Returns error SQL16061N	Query returns: FALSE No error is issued. Because the comparison returns FALSE, no rows are retrieved by the query.

This new behavior might also occur when you use the XMLTABLE function. The XMLTABLE function allows you to execute an XQuery expression and return values as a table instead of as a sequence of values. In the COLUMNS clause of the XMLTABLE function, you define the characteristics of each column, such as the data type. For CHAR and VARCHAR columns, if any non-blank characters are truncated, the XMLTABLE function returns warning SQL0445W.

Optimizer can now choose VARCHAR indexes for queries that contain fn:starts-with

Starting in DB2 V10.1, for queries with predicates that contain the fn:starts-with function, the optimizer can now choose to use VARCHAR-type indexes.

Details

In previous releases, queries with predicates that contained the fn:starts-with function could not use XML indexes for access and had to use table scans. The fn:starts-with function determines whether a string begins with a specific substring.

Resolution

In DB2 V10.1, for queries with predicates that contain fn:starts-with, the optimizer can choose to use VARCHAR-type indexes to speed up the query. No changes are necessary to existing VARCHAR indexes, and you do not need to use any special syntax in the CREATE INDEX statement for new indexes.

For example, consider the index created by the following statement:

```
CREATE INDEX varcharidx ON favorite_cds (cdinfo)
  GENERATE KEYS USING XMLPATTERN
  '/favoritecds/cd/year' as SQL VARCHAR(20);
```

In Version 9.8, and earlier, the following query must perform a table scan, whereas, in DB2 V10.1, the optimizer can choose to use the index **varcharidx**, if it offers a faster query response:

```
XQUERY for $y in db2-fn:xmlcolumn
  ('FAVORITE_CDS.CDINFO')/FAVORITECDS/CD
  [YEAR/fn:starts-with(., "199")] return $y;
```

The use of indexes of type VARCHAR HASHED with these kinds of queries is not supported.

CREATE GLOBAL TEMPORARY TABLE and DECLARE GLOBAL TEMPORARY TABLE statements have been changed

Implicitly hidden columns in created temporary tables and declared temporary tables are not supported.

Details

You can not use the LIKE clause to create created temporary tables or declared temporary tables from a base table that contains implicitly hidden columns. An error is returned. In previous releases, you could create temporary tables with the LIKE clause specifying a base table with implicitly hidden columns.

If you use the AS (*fullselect*) DEFINITION ONLY clause to create created temporary tables or declared temporary tables from a base table that contains implicitly hidden columns, the new columns in the temporary table do not inherit the implicitly hidden attribute.

Resolution

Start using the AS (*fullselect*) DEFINITION ONLY clause instead of the LIKE clause on tables that contain implicitly hidden columns. Keep in mind that the new columns in the temporary tables will not be implicitly hidden columns. You can choose to exclude the implicitly hidden columns in the full select.

Also, modify any scripts or applications using the LIKE clause to create created temporary tables or declared temporary tables from a base table that contains implicitly hidden columns.

Some catalog views and built-in routines changed to include member information

In Version 10.1, some system catalog views and administrative SQL routines changed to include the database member information.

Details

The following system catalog views have added a new MEMBER column:

- SYSCAT.EVENTMONITORS
- SYSCAT.PACKAGES

The following administrative SQL views and routines return a new MEMBER column:

- APPL_PERFORMANCE
- APPLICATIONS¹
- AUDIT_ARCHIVE
- BP_HITRATIO
- BP_READ_IO
- BP_WRITE_IO
- DBCFG
- DB_GET_CFG
- LOG_UTILIZATION

- LONG_RUNNING_SQL
- LOCKS_HELD³
- LOCKWAITS³
- PD_GET_DIAG_HIST
- PD_GET_LOG_MSGS
- PDLOGMSG_LAST24HOURS
- QUERY_PREP_COST
- SNAP_GET_AGENT
- SNAP_GET_AGENT_MEMORY_POOL³
- SNAP_GET_APPL_INFO¹
- SNAP_GET_APPL
- SNAP_GET_BP_PART
- SNAP_GET_BP
- SNAP_GET_DB_MEMORY_POOL³
- SNAP_GET_DB
- SNAP_GET_DBM_MEMORY_POOL³
- SNAP_GET_DBM
- SNAP_GET_DETAILLOG
- SNAP_GET_DYN_SQL
- SNAP_GET_FCM
- SNAP_GET_FCM_PART²
- SNAP_GET_HADR³
- SNAP_GET_LOCK³
- SNAP_GET_LOCKWAIT³
- SNAP_GET_STMT
- SNAP_GET_SUBSECTION
- SNAP_GET_SWITCHES
- SNAP_GET_TAB
- SNAP_GET_TAB_REORG
- SNAP_GET_TBSP
- SNAP_GET_TBSP QUIESCER
- SNAP_GET_UTIL
- SNAP_GET_UTIL_PROGRESS
- SNAP_WRITE_FILE
- SNAPAGENT
- SNAPAGENT_MEMORY_POOL³
- SNAPAPPL
- SNAPAPPL_INFO¹
- SNAPBP
- SNAPBP_PART
- SNAPDB
- SNAPDB_MEMORY_POOL³
- SNAPDBM
- SNAPDBM_MEMORY_POOL³
- SNAPDETAILLOG

- SNAPDYN_SQL
- SNAPFCM
- SNAPFCM_PART²
- SNAPHADR³
- SNAPLOCK³
- SNAPLOCKWAIT³
- SNAPSTMT
- SNAPSUBSECTION
- SNAPSWITCHES
- SNAPTAB
- SNAPTAB_REORG
- SNAPTbsp
- SNAPTbsp QUIESCER
- SNAPUTIL
- SNAPUTIL_PROGRESS
- TOP_DYNAMIC_SQL
- WLM_GET_QUEUE_STATS
- WLM_GET_SERVICE_CLASS_AGENTS
- WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES¹
- WLM_GET_SERVICE_SUBCLASS_STATS
- WLM_GET_SERVICE_SUPERCLASS_STATS
- WLM_GET_WORK_ACTION_SET_STATS
- WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES¹
- WLM_GET_WORKLOAD_STATS

¹ This routine also returns an additional new column called COORD_MEMBER.

² This routine also returns an addition new column called FCM_MEMBER.

³ This routine is deprecated in DB2 Version 10.1.

The following administrative SQL routine returns a NUM_MEMBER column instead of the MEMBER column:

- ENV_INST_INFO

In addition, the ADMIN_GET_DBP_MEM_USAGE table function, which returned the total memory consumption for an instance, is deprecated. The ADMIN_GET_DBP_MEM_USAGE table function is replaced by the ADMIN_GET_MEM_USAGE table function. The differences between the two routines are: the name, and that the new table function returns information about each member in the instance.

Resolution

For these system catalog views and administrative SQL routines, if your applications do not use explicit column names inside SELECT statements, then the new MEMBER column is also returned. This additional return result can have an impact on existing applications.

To use the affected routines in a DB2 pureScale environment, you may need to update your applications to use the new MEMBER column explicitly.

Outside of a DB2 pureScale environment, nothing is required to use the new functionality. The only difference is that the new MEMBER column is returned. You may need to modify your application queries if they use a wildcard or a SELECT * FROM ... code phrase.

If you previously used the ADMIN_GET_DBP_MEM_USAGE table function in your applications, you need to modify your application to use the ADMIN_GET_MEM_USAGE table function instead.

Some built-in string functions have changed result data types in Unicode databases

In Version 10.1, the result data type of some built-in string functions have been changed to align correctly to the detailed documentation topic for the function.

Details

In previous releases, invocations of certain built-in string functions in Unicode databases would incorrectly return a character data type instead of a graphic data type.

The following functions are affected:

- CONCAT
- INSERT
- LISTAGG
- LPAD
- OVERLAY
- REPLACE
- RPAD
- TRANSLATE

For example:

- In previous releases, the following function had a result data type of VARCHAR(10):

```
LPAD(g'abcdefg', 10, ?)
```

In Version 10.1, the result data type is VARGRAPHIC(10).

- In previous releases, the following function had a result data type of VARCHAR(5):

```
RPAD(DBCLOB('abc'), 5)
```

In Version 10.1, the result data type is VARGRAPHIC(5).

- In previous releases, the following function had a result data type of VARCHAR(254):

```
TRANSLATE(?, VARGRAPHIC('abc'), VARGRAPHIC('def'))
```

In Version 10.1, the result data type is VARGRAPHIC(254).

Resolution

If you require that the data type remain unchanged, you can explicitly cast the result.

If you have an existing materialized query table (MQT) that uses an affected invocation of the built-in string functions in the select list of the fullselect used to define the MQT, you could get an error SQLCODE -344 (SQL0344N) when accessing the MQT. To resolve the issue, drop and recreate the affected MQT.

Examples

- `VARCHAR(LPAD(g'abcdefg', 1, ?), 10)` returns a result data type of `VARCHAR(10)`.
- `VARCHAR(RPAD(DBCLOB('abc'), 1), 5)` returns a result data type of `VARCHAR(5)`.
- `VARCHAR(TRANSLATE(?, VARGRAPHIC('abc'), VARGRAPHIC('def')), 254)` returns a result data type of `VARCHAR(254)`.

Some special registers have changed

To support new features in Version 10.1, special registers have been added and modified.

Details

The following special registers have been added:

- `CURRENT TEMPORAL BUSINESS_TIME`
- `CURRENT TEMPORAL SYSTEM_TIME`

Resolution

Start using these new special registers in your applications and scripts. The following statements show how to change their value:

- `SET CURRENT TEMPORAL BUSINESS_TIME`
- `SET CURRENT TEMPORAL SYSTEM_TIME`

Some system catalog views, built-in functions and global variables, built-in administrative routines and views have been added and changed

To support new features in Version 10.1, database catalog objects such as system catalog views, built-in functions and global variables, built-in administrative routines and views have been added and modified.

Details

System catalog view changes

The following system catalog views have changed in Version 10.1. Most modifications to catalog views consist of new columns, changed descriptions, changed column data types, and increased column lengths.

- `SYSCAT.AUDITUSE`
- `SYSCAT.BUFFERPOOLDBPARTITIONS`

- SYSCAT.BUFFERPOOLEXCEPTIONS
- SYSCAT.COLUMNS
- SYSCAT.CONTEXTS
- SYSCAT.DATAPARTITIONS
- SYSCAT.DATATYPES
- SYSCAT.DBAUTH
- SYSCAT.EVENTMONITORS
- SYSCAT.EVENTS
- SYSCAT.EVENTTABLES
- SYSCAT.HISTOGRAMTEMPLATEUSE
- SYSCAT.INDEXCOLUSE
- SYSCAT.INDEXES
- SYSCAT.INDEXXMLPATTERNS
- SYSCAT.INVALIDOBJECTS
- SYSCAT.PACKAGEDEP
- SYSCAT.PACKAGES
- SYSCAT.ROLES
- SYSCAT.ROUTINES
- SYSCAT.ROWFIELDS
- SYSCAT.SCHEMATA
- SYSCAT.SERVICECLASSES
- SYSCAT.TABCONST
- SYSCAT.TABLES
- SYSCAT.TABLESPACES
- SYSCAT.THRESHOLDS
- SYSCAT.TRANSFORMS
- SYSCAT.TRIGGERS
- SYSCAT.VARIABLES
- SYSCAT.WORKACTIONS
- SYSCAT.WORKLOADS

The following system catalog views have been added in Version 10.1:

- SYSCAT.CONTROLDEP
- SYSCAT.CONTROLS
- SYSCAT.PERIODS
- SYSCAT.SCPREFTBSPACES
- SYSCAT.STATEMENTTEXTS
- SYSCAT.STOGROUPS
- SYSCAT.USAGELISTS
- SYSCAT.WORKCLASSATTRIBUTES

The following system catalog views have been deprecated in Version 10.1:

- SYSCAT.BUFFERPOOLDBPARTITIONS. Start using the SYSCAT.BUFFERPOOLEXCEPTIONS catalog view instead.

Built-in function changes

The following built-in functions have changed in Version 10.1:

- TIMESTAMPDIFF scalar function

The following built-in functions have been added:

- INSTRB scalar function
- TO_SINGLE_BYTE scalar function
- VERIFY_GROUP_FOR_USER scalar function
- VERIFY_ROLE_FOR_USER scalar function
- VERIFY_TRUSTED_CONTEXT_ROLE_FOR_USER scalar function

Built-in global variable changes

The following built-in global variables have been added:

- CLIENT_HOST
- CLIENT_IPADDR
- CLIENT_ORIGUSERID
- CLIENT_USRSECTOKEN
- MON_INTERVAL_ID
- PACKAGE_NAME
- PACKAGE_SCHEMA
- PACKAGE_VERSION
- ROUTINE_MODULE
- ROUTINE_SCHEMA
- ROUTINE_SPECIFIC_NAME
- ROUTINE_TYPE
- TRUSTED_CONTEXT

Built-in administrative views and routine changes

The following administrative views and routines have changed in Version 10.1. Most modifications consist of new columns, new values, changed column data types, and increased column lengths:

- ADMIN_GET_INDEX_INFO table function
- APPLICATIONS administrative view
- APPL_PERFORMANCE administrative view
- BP_HITRATIO administrative view
- BP_READ_IO administrative view
- BP_WRITE_IO administrative view
- ENV_GET_SYSTEM_RESOURCES
- ENV_SYS_RESOURCES administrative view
- EVMON_FORMAT_UE_TO_TABLES procedure
- LOG_UTILIZATION administrative view
- LONG_RUNNING_SQL administrative view
- MON_BP_UTILIZATION administrative view
- MON_FORMAT_XML_METRICS_BY_ROW table function
- MON_GET_ACTIVITY_DETAILS table function
- MON_GET_BUFFERPOOL table function
- MON_GET_CONNECTION table function
- MON_GET_CONNECTION_DETAILS table function
- MON_GET_CONTAINER table function

- MON_GET_FCM
- MON_GET_FCM_CONNECTION_LIST
- MON_GET_INDEX table function
- MON_GET_PAGE_ACCESS_INFO table function
- MON_GET_PKG_CACHE_STMT table function
- MON_GET_PKG_CACHE_STMT_DETAILS table function
- MON_GET_SERVICE_SUBCLASS table function
- MON_GET_SERVICE_SUBCLASS_DETAILS table function
- MON_GET_TABLE table function
- MON_GET_TABLESPACE table function
- MON_GET_UNIT_OF_WORK table function
- MON_GET_UNIT_OF_WORK_DETAILS table function
- MON_GET_WORKLOAD table function
- MON_GET_WORKLOAD_DETAILS table function
- PDLOGMSGS_LAST24HOURS administrative view
- QUERY_PREP_COST administrative view
- SNAPAGENT administrative view
- SNAPAGENT_MEMORY_POOL administrative view
- SNAP_GET_AGENT table function
- SNAP_GET_BP_PART table function
- SNAP_GET_DBM_MEMORY_POOL table function
- SNAP_GET_DB_MEMORY_POOL table function
- SNAP_GET_FCM table function
- SNAP_GET_FCM_PART table function
- SNAP_GET_LOCK table function
- SNAP_GET_LOCKWAIT table function
- SNAP_GET_STMT table function
- SNAP_GET_SUBSECTION table function
- SNAP_GET_SWITCHES table function
- SNAP_GET_TBSP QUIESCER table function
- SNAP_GET_UTIL table function
- SNAP_GET_UTIL_PROGRESS table function
- SYSINSTALLOBJECTS procedure
- WLM_GET_SERVICE_SUPERCLASS_STATS table function
- WLM_GET_WORK_ACTION_SET_STATS table function

The following ADMIN_CMD stored procedure and associated administrative SQL routines have been added:

- ADMIN_GET_INTRA_PARALLEL scalar function
- ADMIN_GET_STORAGE_PATHS table function
- ADMIN_GET_TAB_COMPRESS_INFO table function
- ADMIN_GET_TAB_DICTIONARY_INFO table function
- ADMIN_GET_TAB_INFO table function
- ADMIN_SET_INTRA_PARALLEL procedure

The following text search procedures have been added or changed:

- SYSTS_CLEANUP procedure

- SYSTS_CONFIGURE procedure
- SYSTS_DROP procedure
- SYSTS_ENABLE procedure
- SYSTS_UPDATE procedure
- SYSTS_UPGRADE_CATALOG procedure
- SYSTS_UPGRADE_INDEX procedure

The following monitor routines have been added:

- MON_GET_APPLICATION_HANDLE scalar function
- MON_GET_APPLICATION_ID scalar function
- MON_GET_AUTO_MAINT_QUEUE table function
- MON_GET_AUTO_RUNSTATS_QUEUE table function
- MON_GET_CF table function
- MON_GET_CF_CMD table function
- MON_GET_CF_WAIT_TIME table function
- MON_GET_EXTENDED_LATCH_WAIT table function
- MON_GET_GROUP_BUFFERPOOL table function
- MON_GET_HADR table function
- MON_GET_INDEX_USAGE_LIST table function
- MON_GET_REBALANCE_STATUS table function
- MON_GET_RTS_RQST table function
- MON_GET_TABLE_USAGE_LIST table function
- MON_GET_TRANSACTION_LOG table function
- MON_GET_USAGE_LIST_STATUS
- MON_INCREMENT_INTERVAL_ID stored procedure
- MON_SAMPLE_SERVICE_CLASS_METRICS table function
- MON_SAMPLE_WORKLOAD_METRICS table function

The following environment routines have been added:

- ENV_GET_DB2_SYSTEM_RESOURCES table function
- ENV_GET_NETWORK_RESOURCES table function

The following snapshot routines and views have been added:

- SNAP_GET_APPL_INFO table function
- SNAP_GET_APPL table function
- SNAP_GET_BP table function
- SNAP_GET_CONTAINER table function
- SNAP_GET_DB table function
- SNAP_GET_DBM table function
- SNAP_GET_DETAILLOG table function
- SNAP_GET_DYN_SQL table function
- SNAP_GET_TAB table function
- SNAP_GET_TBSP_PART table function
- SNAP_GET_TBSP table function

The following workload management routines have been added:

- WLM_GET_SERVICE_CLASS_AGENTS table function

- WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES table function
- WLM_GET_SERVICE_SUBCLASS_STATS table function
- WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES table function
- WLM_GET_WORKLOAD_STATS table function

Resolution

Adjust your applications to the changes in existing system catalog objects such as new columns, or modified data type in columns. Also, start using new system catalog objects that provide more comprehensive views or routines, or access to information about new functionality.

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. To minimize the impact of changes to built-in routines and views, review “Best practices for calling built-in routines and views in applications” in *Administrative Routines and Views* .

For a list of the data dictionary-compatible views, see the “Data dictionary-compatible views” topic.

Chapter 20. DB2 command and SQL statement changes summary

Version 10.1 introduces changes to DB2 CLP commands, DB2 system commands, and SQL statements to support new capabilities. These changes can affect your existing database applications or database administration scripts.

The following DB2 system commands have been changed:

- The **db2cat** command (see “db2cat command output text has changed”)
- The **db2ckupgrade** command (see “db2ckupgrade command enhanced for partitioned environments” on page 132)
- The **db2cluster_prepare** command (see “Some parameters for commands that manage instances have changed” on page 132)
- The **db2evtbl** command (see “db2evtbl command has been changed” on page 133)
- The **db2exfmt** command (see “db2exfmt command output has changed for partitioned tables” on page 134)
- The **db2icrt** command (see “Some parameters for commands that manage instances have changed” on page 132)
- The **db2iupdt** command (see “Some parameters for commands that manage instances have changed” on page 132)
- The **db2pd** command (see “db2pd command has changed to support new functionality” on page 134)

The following SQL statements have been changed:

- The ALTER TABLE statement (see “ALTER TABLE and CREATE TABLE statement have been changed” on page 135)
- The ALTER TABLESPACE statement (see “ALTER TABLESPACE statement has a new clause” on page 136)
- The ALTER WORKLOAD statement (see “ALTER WORKLOAD and CREATE WORKLOAD statements have changed” on page 137)
- The CREATE INDEX statement (see “DETAILED keyword in CREATE INDEX statement now indicates a changed default behavior” on page 138)
- The CREATE TABLE statement (see “ALTER TABLE and CREATE TABLE statement have been changed” on page 135)
- The CREATE WORKLOAD statement (see “ALTER WORKLOAD and CREATE WORKLOAD statements have changed” on page 137)

Also, review DB2 CLP commands, DB2 system commands, and SQL statements that are deprecated or discontinued because they can also affect your existing database applications or database administration scripts.

db2cat command output text has changed

The output for the **db2cat** command no longer includes properties for the table space where the database object is stored.

Details

Due to changes for storage groups in Version 10.1, the following table space properties are no longer reported by the **db2cat** command:

- TRANSFERRATE
- OVERHEAD
- WRITETRANSFERRATE
- WRITEOVERHEAD
- PREFETCHSIZE
- EXTENTSIZE
- PAGESIZE

The SYSCAT.TABLESPACES catalog view provides the values for all of these table space properties. The EXPLAIN_ARGUMENT table also provides information about the values used by the optimizer for the TRANSFERRATE, OVERHEAD, and PREFETCHSIZE properties.

In previous releases, the **db2cat** command included table space properties along with the packed descriptor information for tables.

Resolution

To determine the values for the table space properties, first determine the table space where the table or index is stored and then query the SYSCAT.TABLESPACES catalog view for that table space.

db2ckupgrade command enhanced for partitioned environments

The **db2ckupgrade** command now verifies that all database partitions are ready for upgrade by default. In previous releases, it verified only the current database partition.

Details

In previous releases, you had to run the **db2ckupgrade** command on each database partition to verify all database partitions. In Version 10.1 you can run once from any partition and it will verify all database partitions.

Resolution

Issue the **db2ckupgrade** command once to check all database partitions to ensure a successful instance upgrade.

Make sure that the **db2ckupgrade** command completes successfully before you issue the **db2iupgrade** command.

Some parameters for commands that manage instances have changed

The **db2icrt**, **db2iupdt**, **db2cluster_prepare** commands have new parameters, changes to existing parameters, and deprecation of parameter value format.

Details

In Version 10.1, the **-m** and **-cf** parameters in the **db2icrt**, **db2iupdt**, and **db2cluster_prepare** commands indicate the hostname of a member or a cluster caching facility (CF) respectively. The cluster interconnect netname is the hostname of the interconnect used for high speed communication between members and cluster caching facilities. To specify the cluster interconnect netname for members, use the new **-mnet** parameter. To specify the cluster interconnect netname for CFs, use the new **-cfnet** parameter.

In previous releases, you could use the **-m** and **-cf** parameters to indicate the hostname and the network name using a colon as separator. This format for these parameters is deprecated and might be discontinued in a future release.

If you want to indicate an IPv6 IP address for the hostname, you must indicate the hostname with the **-m** and **-cf** parameters, and the network name with the **-mnet** and **-cfnet** parameters. Using the old format available in previous releases to indicate both hostname and network name will return an error.

In addition, you can use the **instance_shared_mount** parameter in Version 10.1 to indicate the directory where you want to mount a newly created DB2 managed clustered file system. If you do not use this parameter, the database manager generates a name.

Resolution

Start using the new **-mnet** and **-cfnet** parameters to indicate the network name before the old format becomes discontinued. Modify any existing scripts and applications to use the new parameters.

Start using the **instance_shared_mount** parameter to indicate a directory to mount the DB2 managed clustered file system instead of having a system-generated name.

db2evtbl command has been changed

This command can now generate DDL to create change history event monitors. The command also now generates DDL to create the locking, unit of work and package cache event monitors with the new default target type of TABLE.

Details

In previous releases, the locking, package cache, and unit of work event monitors could write only to unformatted event (UE) tables. As a result, the **db2evtbl** command generated CREATE TO EVENT MONITOR statements with the WRITE TO UNFORMATTED EVENT TABLE clause. Starting in Version 10.1, the locking, package cache, and unit of work event monitors also support a target of TABLE, in addition to UNFORMATTED EVENT TABLE. The **db2evtbl** command now generates CREATE TO EVENT MONITOR statements with the WRITE TO TABLE clause instead of the WRITE TO UNFORMATTED EVENT TABLE clause.

Resolution

Use the **db2evtbl** command to generate the CREATE TO EVENT MONITOR statements with the WRITE TO TABLE clause, including for the new change history event monitor.

If you require DDL statements to create locking, unit or work or package cache event monitors that write to UE tables, you can use the **db2evtbl** command to generate the CREATE TO EVENT MONITOR statements with the WRITE TO TABLE clause. Then, edit the generated statements and replace “WRITE TO TABLE” with “WRITE TO UNFORMATTED EVENT TABLE”.

db2exfmt command output has changed for partitioned tables

The **db2exfmt** command output now shows the table space attributes value for each table space containing a partitioned table.

Details

In Version 10.1, the EXPLAIN_OBJECT table contains a single value for each table space in which a partitioned table exists.

In previous releases, for a partitioned table, the table space attributes value that was stored in the EXPLAIN_OBJECT table was the aggregated table space attribute value stored in all corresponding table spaces. If you ran the **db2exfmt** command to format the EXPLAIN_OBJECT table, the output used to print the aggregate value of table space attributes for partitioned tables.

For statements that you compile with the CURRENT EXPLAIN MODE set to YES in Version 10.1, if a table space attribute value for a partitioned table is identical for each table space containing the partitioned table, the EXPLAIN_OBJECT table contains that value. Otherwise, if the attributes differ, the EXPLAIN_OBJECT table contains the value -1.

For example, if the table space attributes OVERHEAD, TRANSFERRATE and PREFETCHSIZE differ, then the respective OVERHEAD, TRANSFERRATE and PREFETCHSIZE column in the EXPLAIN_OBJECT table contains the value -1 to indicate that the corresponding attribute is different among the various table spaces.

Stored procedures read the sections and write the explain information into the explain tables. For partitioned tables or indexes, the following procedures write the value for the attributes into the EXPLAIN_OBJECT table:

- EXPLAIN_FROM_ACTIVITY
- EXPLAIN_FROM_CATALOG
- EXPLAIN_FROM_DATA
- EXPLAIN_FROM_SECTION

Resolution

Adjust applications or scripts that use the **db2exfmt** command to process the changed output for partitioned tables.

db2pd command has changed to support new functionality

The new **-storagegroups** parameter has been added in support of storage groups. Also, the output report for the **-catalogcache**, **-hadr**, **-serviceclasses**, **-storagepaths**, **-tablespaces**, **-tcbstats**, **-workclasssets**, and parameters have changed to support new functionality.

Details

In Version 10.1, the new **-storagegroups** parameter provides details about storage groups and storage paths. If you do not specify a storage group identifier, information about all storage groups is reported. The **-storagepaths** parameter continues to report information about all storage groups.

Also, the output reports for the following parameters have been changed:

- The **-catalogcache** parameter output reports details about the new table space cache.
- The **-hadr** parameter output has been completely changed. Existing fields have been renamed to match the output from the new MON_GET_HADR table function, and fields have been added to support monitoring for the new HADR multiple standby. The formatting of the output has also changed from multiple field names on one line and field values on the subsequent line to having one field per line.
- The **serviceclasses** parameter output now includes the UOW throughput, activity throughput, activity lifetime average, CPU Velocity, and CPU Utilization.
- The **-storagepaths** parameter output includes storage group configuration details.
- The **-tablespaces** parameter output now includes the storage group identifier that a table space is associated with and table space storage statistics.
- The **-tcbstats** parameter output for the RowsComp and RowsUNcomp columns has been removed. Also, the output for the column and data for IndexObjSize has been deprecated. If you perform a reorganization to reclaim extents, the IndexObjSize output does not accurately reflect *the number of pages in the index object* because the value still includes the pages that were released during the reorganization. You should use instead the INDEX_OBJECT_P_SIZE or INDEX_OBJECT_L_SIZE columns of the ADMIN_GET_INDEX_INFO table function to obtain accurate values.
- The **-workclasssets** parameter output is no longer in a table format. The output format lists basic work class information followed by the work class attribute information.

Resolution

Change applications or scripts that use the output of the **db2pd** command to adjust to these changes.

Start using SQL administrative routines and views that report the same information instead of the command output as they offer more flexibility for programming.

ALTER TABLE and CREATE TABLE statement have been changed

The COMPRESS clause for the ALTER TABLE and CREATE TABLE statements have a new format and the default compression method has changed. Also, the IMPLICITLY HIDDEN clause can be used in any type of column.

Details

Version 10.1 introduces a new method of compressing data rows that use page-level compression dictionaries and a table-level compression dictionary called

adaptive row compression. In previous releases, the method of compressing data rows used a table-level compression dictionary and it is now called *classic row compression*.

The COMPRESS YES ADAPTIVE or COMPRESS YES clauses on the CREATE TABLE and ALTER TABLE statements enable adaptive row compression on a table. The new COMPRESS YES STATIC clause enables classic row compression. The COMPRESS YES clause continues to be the default. However, it has a different behavior than in previous releases because it enables adaptive row compression. In previous releases, the COMPRESS YES clause enabled classic row compression.

After enabling row compression on an existing table, write operations only result in compressed data rows. Existing data on the table remains in uncompressed format. To compress the existing data, you must reorganize the table or reload the data to rebuild the page-level compression dictionaries and table-level compression dictionary.

In Version 10.1, the IMPLICITLY HIDDEN clause in the ALTER TABLE and CREATE TABLE statements can be used in any type of column. In previous releases, you can only specify this clause for ROW CHANGE TIMESTAMP columns. For more details on tables that use IMPLICITLY HIDDEN columns, see “Time period and transaction start columns for system-period temporal tables”.

Resolution

If you have scripts or applications that issue the ALTER TABLE or CREATE TABLE statements with the COMPRESS YES clause, make sure that you add the STATIC or ADAPTIVE keyword to explicitly indicate the table compression method that you want.

After upgrading your databases from previous releases, to enable adaptive row compression on existing tables, issue the ALTER TABLE statement with the COMPRESS YES ADAPTIVE clause and rebuild the compression dictionaries. For more details, see “Improved table compression rates with greater ease of use”. To continue to enable classic row compression, you must use the COMPRESS YES STATIC clause.

Start using hidden columns in new and existing tables. To add hidden columns to existing tables, use the **LOAD**, **IMPORT**, or **EXPORT** commands with the `includeimplicitlyhidden` modifier to populate them.

ALTER TABLESPACE statement has a new clause

You can now explicitly suspend a rebalance operation that is in progress during performance-sensitive periods and resume at a later time.

Details

To suspend the rebalance operation, issue the ALTER TABLESPACE statement with the REBALANCE SUSPEND clause. This places the operation into suspended state.

To resume the operation, issue the ALTER TABLESPACE statement with the REBALANCE RESUME clause.

The suspended state is persistent and the rebalance operation is restarted upon database activation.

You can monitor rebalance operations in progress using the `MON_GET_REBALANCE_STATUS` table function.

Resolution

Start using the `ALTER TABLESPACE` statement to suspend or resume a rebalance operation.

ALTER WORKLOAD and CREATE WORKLOAD statements have changed

The syntax for the `ALTER WORKLOAD` and `CREATE WORKLOAD` statements has changed to support new functionality such as enabling application intrapartition parallelism, limiting the maximum runtime degree for an application, and collecting information.

Details

Enabling application intrapartition parallelism and limiting the maximum runtime degree for an application

When you create or modify workloads, you can use the new `MAXIMUM DEGREE` clause for the following purposes:

- To enable or disable intrapartition parallelism for applications that you assign to the workload
- To limit the system resource that an application can consume

Collecting information

You can now specify two new values for the `BASE` option of the `COLLECT UNIT OF WORK DATA` clause: `INCLUDE PACKAGE LIST` and `INCLUDE EXECUTABLE LIST`. You can use these new values to indicate whether you want to collect information about the executable ID list, package list, or both for transactions that are associated with the workload. This information is sent to the unit of work event monitors. In previous releases, you could collect only package list information for transactions that were associated with a workload, by using the `PACKAGE LIST` option for the `COLLECT UNIT OF WORK DATA` clause.

Resolution

Enabling application intrapartition parallelism and limiting the maximum runtime degree for an application

Start using the new `MAXIMUM DEGREE` clause to enable or disable intrapartition parallelism for applications that you assign to a workload or limit the maximum runtime degree of an application.

Collecting information

Modify scripts and applications that use the `COLLECT UNIT OF WORK DATA PACKAGE LIST` clause. Use the new syntax to collect package list information, as shown in the following example:

```
ALTER WORKLOAD REPORTS COLLECT UNIT OF WORK DATA BASE
    INCLUDE PACKAGE LIST
```

If you want to collect information about the executable ID list, use the new syntax that is shown in the following example:

```
ALTER WORKLOAD REPORTS COLLECT UNIT OF WORK DATA BASE
    INCLUDE PACKAGE LIST, EXECUTABLE LIST
```


Although the COLLECT UNIT OF WORK DATA PACKAGE LIST clause is still supported for compatibility with previous releases, this syntax is nonstandard, and you should not use it.

DETAILED keyword in CREATE INDEX statement now indicates a changed default behavior

The DETAILED keyword in the CREATE INDEX statement now specifies that a sampling technique is to be used when processing index entries to collect extended index statistics.

Details

In DB2 Version 10.1, specifying the DETAILED keyword in the CREATE INDEX statement is the same as specifying the SAMPLED DETAILED keyword in the CREATE INDEX statement.

In previous releases, the DETAILED keyword in the CREATE INDEX statement specified that all the index entry were to be examined individually.

To maintain the same behavior as in previous releases, use the UNSAMPLED DETAILED keyword with the CREATE INDEX statement.

Resolution

Use the SAMPLED DETAILED or UNSAMPLED DETAILED keywords when you issue a CREATE INDEX statement to get the wanted behavior.

Chapter 21. Deprecated functionality

Functionality gets marked as *deprecated* when a specific function or feature is supported in the current release but might be removed in a future release. In some cases, it might be advisable to plan to discontinue the use of deprecated functionality.

For example, a registry variable might be deprecated in this release because the behavior triggered by the registry variable has been enabled by default in this release, and the obsolete registry variable will be removed in a future release.

DB2 functionality

The following DB2 functionality has been deprecated:

- Activity monitor routines (see “Activity monitor routines have been deprecated” on page 140)
- Agent priority of service classes (see “Agent priority of service classes has been deprecated” on page 141)
- Collations based on the Unicode Standard version 4.00 (see “Collations based on the Unicode Collation Algorithm of the Unicode Standard version 4.0.0 have been deprecated” on page 141)
- INSTALL_TSAMP response file keyword (see “IBM Tivoli System Automation for Multiplatforms (SA MP) is now automatically installed” on page 112)
- SMS user table spaces (see “SMS permanent table spaces have been deprecated” on page 142)
- Automatic statistics profiling (see “Automatic statistics profiling is deprecated” on page 143)
- Some version-suffixed SQL administrative routines (see “Some version-suffixed SQL administrative routines have been deprecated” on page 144)
- SNAPHADR administrative view and SNAP_GET_HADR table function (see “Some monitoring interfaces for HADR have been deprecated” on page 153)

DB2 database products

The following product has been deprecated:

- Net Search Extender (see “Net Search Extender has been deprecated” on page 145)

Commands, command parameters, statements, registry variables, configuration parameters, and monitor elements

The following commands, command parameters, statements, registry variables, configuration parameters, and monitor elements have been deprecated:

- The **db2IdentifyType1** command (see “db2IdentifyType1 command has been deprecated” on page 145)
- The **db2_install** command (see “db2_install command has been deprecated (Linux and UNIX)” on page 146)
- The **dynexpln** command (see “dynexpln command has been deprecated” on page 146)

- The **PRUNE LOGFILE** command (see “PRUNE LOGFILE command has been deprecated” on page 147)
- Some parameters of the **CREATE DATABASE** command (see “Some parameters of the CREATE DATABASE command have been deprecated” on page 148)
- The format to indicate both hostname and network name with the **-m** and **-cf** parameters of the **db2icrt**, **db2iupdt**, and **db2cluster_prepare** commands
The format to indicate both hostname and network name with the **-m** and **-cf** parameters in some commands that manage instances (see “Some parameters for commands that manage instances have changed” on page 132)
- The **-flushbp** parameter of the **db2pdcfg** command (see “-flushbp parameter of the db2pdcfg command has been deprecated” on page 149)
- One of the columns output by the **-tcbstat** parameter of the **db2pd** command (see “**db2pd** command has changed to support new functionality” on page 134)
- The **ONLY** keyword in the **CLEANUP ONLY** and **RECLAIM EXTENTS ONLY** parameters of the **REORG INDEXES/TABLE** command (see “REORG INDEXES/TABLE command parameters and parameter values for related DB2 API data structures have been deprecated or discontinued” on page 150)
- The **ALTER DATABASE** statement (see “ALTER DATABASE statement has been deprecated” on page 151)
- Some registry and environment variables (see “Some registry and environment variables have been deprecated” on page 151)
- The **ENHANCED_MULTIPLE_DISTINCT** setting for the **DB2_EXTENDED_OPTIMIZATION** registry variable (see “Some registry and environment variables have changed” on page 107)
- Some database configuration parameter (see “Some database configuration parameters have been deprecated or discontinued” on page 152)
- Some monitor elements for HADR (see “Some monitoring interfaces for HADR have been deprecated” on page 153)
- The **-global** option for troubleshooting tools
“The -global option for troubleshooting tools has been deprecated” on page 154

Review each topic to find out more details and to plan for future changes. Review also Chapter 22, “Discontinued functionality,” on page 155 that might affect your databases and existing applications.

Activity monitor routines have been deprecated

Activity monitor routines have been deprecated and might be removed in a future release.

Details

Activity monitor was a graphical reporting tool that was part of the Control Center tools to provide monitor reports. In addition, a set of activity monitor routines provided predefined reports based on a specific subset of monitor data.

In Version 10.1, the DB2 Control Center tools have been discontinued. Therefore, the activity monitor GUI is no longer available. As a result, the following activity monitor routines have been deprecated:

- AM_BASE_RPT_RECOMS
- AM_BASE_RPTS
- AM_DROP_TASK
- AM_GET_LOCK_CHN_TB
- AM_GET_LOCK_CHNS
- AM_GET_LOCK_RPT
- AM_GET_RPT
- AM_SAVE_TASK

Resolution

Start to use monitor routines, the **db2pd** command, or IBM InfoSphere® Optim™ tools for activity monitoring as they provide the same information.

Agent priority of service classes has been deprecated

Associating each DB2 service class with an agent priority, which controls the relative operating system priority of agents in the service class, has been deprecated and might be removed in a future release.

Details

In Version 10.1, you can use the workload manager (WLM) dispatcher to control CPU consumption more effectively and allow higher priority work to consume more CPU resources than lower priority work. The WLM dispatcher provides additional features over agent priority.

In Version 9.5, the agent priority was introduced to assign less system CPU resources to lower priority work and more system CPU resources to higher priority work based on the agent priority of the service class the work ran in. However, this method proved to be only effective for certain types of workloads.

For AIX and Linux operating systems, you can also use the integration between DB2 service classes and AIX WLM or Linux WLM classes to control the amount of system resources allocated to each service class.

Resolution

Start to use the WLM dispatcher in place of the agent priority to control CPU consumption.

Collations based on the Unicode Collation Algorithm of the Unicode Standard version 4.0.0 have been deprecated

The UCA400_NO, UCA400_LSK, and UCA400_LTH keywords have been deprecated for the **COLLATE USING** parameter in the **CREATE DATABASE** command.

Details

When you create Unicode databases, you can specify locale-sensitive collations based on the Unicode Collation Algorithm (UCA) with the **COLLATE USING** parameter in the **CREATE DATABASE** command. These collations provide similar functionality and better performance

Resolution

For new Unicode databases, create them using any of the supported locale-sensitive UCA-based collations. See “Unicode Collation Algorithm based collations” for details.

For existing Unicode databases, you can convert to a supported locale-sensitive UCA-based collation by recreating your databases following a similar procedure described in the “Converting non-Unicode databases to Unicode” task.

IBM Tivoli System Automation for Multiplatforms (SA MP) is now automatically installed

IBM Tivoli System Automation for Multiplatforms (SA MP) is now automatically installed when SA MP is a required component for other components selected for installation.

Details

If you select any component that requires SA MP for installation, SA MP is automatically installed whether SA MP is selected or not. For example, IBM DB2 pureScale Feature requires SA MP. If you install DB2 pureScale Feature, SA MP is automatically installed.

As a result, the `INSTALL_TSAMP` response file keyword for DB2 installation is deprecated. If you indicate `INSTALL_TSAMP=NO` in the response file, SA MP is not installed only if any other selected components for installation do not require SA MP.

Resolution

Remove the `INSTALL_TSAMP` keyword from response files. SA MP is only installed when required by other components selected for installation.

SMS permanent table spaces have been deprecated

The system managed spaces (SMS) table space type is now deprecated for permanent table spaces that are defined by the user.

Details

You can still specify the SMS type for catalog table spaces and temporary table spaces. Automatic storage continues to use SMS type for temporary table spaces. The recommended table space types for user table spaces are automatic storage or database managed spaces (DMS).

In previous releases, SMS permanent table spaces were used because they were simple to create and manage. To create a SMS table spaces, you do not have to

specify an initial size but you must ensure that there is enough free disk space. The size and growth of the container files are managed at the operating system level. However, SMS table spaces do not perform as well as DMS table spaces.

With the introduction of automatic storage, management of DMS table spaces was simplified by providing a function that automatically resizes containers. IBM continues to invest and develop in automatic storage and DMS table spaces.

Resolution

For new databases, create them with user table spaces of DMS or automatic storage table space type by using the `CREATE TABLESPACE` statement or the `CREATE DATABASE` command.

For existing SMS permanent table spaces, start converting them to DMS or automatic storage table spaces before SMS user table spaces become discontinued. You can use the `db2move` command or the `LOAD` command specifying the `CURSOR` file type to move tables from the SMS table space to an automatic storage or DMS table spaces.

Automatic statistics profiling is deprecated

Automatic statistics profiling is deprecated and might be removed in a future release.

Details

Automatic statistics profiling is deprecated because of its noticeable performance overhead and restrictions. It is not supported in partitioned database environments, federated systems, or databases with intrapartition parallelism enabled or enables collection of section actuals.

The `ASP` value as a tool name parameter for the `SYSINSTALLOBJECTS` procedure is also deprecated.

In earlier releases, you could use automatic statistics profiling to determine recommended parameters for the `RUNSTATS` command. Also, automatic statistics profiling could detect whether table statistics were outdated.

Statistics profiles for the `RUNSTATS` command are not affected by this deprecation and are still fully supported.

Resolution

Disable the use of automatic statistics profiling by setting the `auto_stats_prof` database configuration parameter to `OFF` before automatic statistics profiling is discontinued.

You can use the IBM Data Studio tool to help you determine the best parameters for the `RUNSTATS` command to continue collecting statistics by using statistics profiles or by running the `RUNSTATS` command. The following capabilities can help you get recommendations for statistics collection:

- Statistics Advisor. For more details, see [Generating and acting on recommendations for collecting statistics for database objects that are in the](#)

access path for an SQL statement at <http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.qrytune.sngqry.doc/topics/genrecsstats.html>.

- Workload Advisor. This advisor requires an active license for IBM InfoSphere Optim Query Workload Tuner tool. For more details, see Generating and acting on recommendations for materialized query tables, multidimensional clustering, and redistributing data across database partitions for query workloads that run on DB2 for Linux, UNIX, and Windows at <http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.qrytune.workloadtunedb2luw.doc/topics/genrecsdsgn.html>.
- Task assistant for the RUNSTATS command. For more details, see Database administration commands that you can run from task assistants at http://publib.boulder.ibm.com/infocenter/dstudio/v3r1/topic/com.ibm.datatools.adm.doc/topics/c_taskassitantcommandsupport.html.

Some version-suffixed SQL administrative routines have been deprecated

A number of version-suffixed SQL administrative routines have been deprecated in Version 10.1. Start using the replacement routines before the deprecated routines are removed in a future release.

Details

Starting with Version 10.1, the names of administrative routines will not have a version suffix to keep the names consistent across releases.

The deprecated SQL administrative routines are as follows:

- ADMIN_GET_TAB_COMPRESS_INFO_V97
- ADMIN_GET_TAB_INFO_V97
- SNAP_GET_APPL_INFO_V95
- SNAP_GET_APPL_V95
- SNAP_GET_BP_V95
- SNAP_GET_CONTAINER_V91
- SNAP_GET_DBM_V95
- SNAP_GET_DB_V97
- SNAP_GET_DETAILLOG_V91
- SNAP_GET_DYN_SQL_V95
- SNAP_GET_STORAGE_PATHS_V97
- SNAP_GET_TAB_V91
- SNAP_GET_TBSP_PART_V97
- SNAP_GET_TBSP_V91
- WLM_GET_SERVICE_CLASS_AGENTS_V97
- WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES_V97
- WLM_GET_SERVICE_SUBCLASS_STATS_V97
- WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES_V97
- WLM_GET_WORKLOAD_STATS_V97

Resolution

Modify your applications and scripts to use the replacement routines or views.

Use good practices, such as the following ones, when issuing queries on SQL administrative routines:

- Use a list of columns rather than the wildcard character so that the query will always return the same number of columns.
- Validate values returned from a column to receive only expected values. For example, verify that the value is within a range of valid values for that column.

Net Search Extender has been deprecated

Net Search Extender (NSE) and all related functionality has been deprecated and might be removed in a future release.

Details

DB2 Text Search is the strategic search solution for DB2 database products.

DB2 Text Search has an architecture that can easily adopt new features and provides the following key functions that sets it apart from NSE:

- Support for linguistic processing
- XQuery, XPath-like search syntax
- Integrated text search server installation
- Stored procedures for index administration
- Support for rich text document formats using the DB2 Accessories Suite package

DB2 Text Search is supported on all operating systems that NSE is supported, except for Linux on System z[®] (64-bit) operating systems. DB2 Text Search provides equivalent functionality to NSE functionality as described in the feature comparison of both solutions.

Resolution

Start using DB2 Text Search instead of NSE before it becomes discontinued. DB2 Text Search allows you to issue SQL and XQuery statements to perform text search queries on data stored in a DB2 database.

For IBM Business Partners applications that use NSE, contact your IBM Business Partner for details about migration from NSE to DB2 Text Search.

For customers that use NSE features that have equivalent features in DB2 Text Search, see “Migration to DB2 Text Search” for details.

db2IdentifyType1 command has been deprecated

The **db2IdentifyType1** command has been deprecated and might be removed in a future release.

Details

In Version 9.7, the **db2IdentifyType1** command was provided to help you convert type-1 indexes to type-2 before upgrading to Version 9.7.

Type-1 indexes were discontinued in Version 9.7. During the database upgrade, type-1 indexes were marked as invalid and automatically rebuilt after the first access to a database or database restart. Therefore, converting type-1 indexes before the upgrade eliminated the overhead of the automatic rebuild.

In Version 10.1, you would use the **db2IdentifyType1** command only before upgrading Version 9.5 databases. Version 9.7 databases no longer have type-1 indexes.

Resolution

If you are upgrading from Version 9.5 to Version 10.1, use this command to generate a script to help you convert your type-1 indexes to type-2 before the upgrade. For details on how to use this command, see “Converting type-1 indexes to type-2 indexes” in *Upgrading to DB2 Version 10.1*.

db2_install command has been deprecated (Linux and UNIX)

The **db2_install** command has been deprecated and might be removed in a future release. Start using the **db2setup** command instead.

Details

The **db2_install** command has been deprecated because it provides the same functionality as the **db2setup** command with a response file. In addition, the **db2setup** command allows you to create and configure instances during the installation process.

Resolution

If you issue the **db2_install** command, a warning message will be returned indicating that the command is deprecated and then it proceeds with the installation as in previous releases.

Although the **db2_install** command is still supported, start using the **db2setup** command with a response file to do silent installs before the **db2_install** command becomes discontinued. Also, change references to the **db2_install** command in installation scripts to use the **db2setup** command with a response file.

If you want the **db2setup** command behave the same as the **db2_install** command, issue the **db2setup** command with a response file that includes following keywords:

```
PROD = ENTERPRISE_SERVER_EDITION
FILE = /opt/ibm/db2/dirname
LIC_AGREEMENT = ACCEPT
INSTALL_TYPE = COMPLETE
```

dynexpln command has been deprecated

The **dynexpln** command has been deprecated and might be removed in a future release.

Details

The **db2expln** command describes the access plan selected for SQL and XQuery statements. The *dynamic-options* of the **db2expln** command allow you to perform all of the functions of the **dynexpln** command.

When you use the *dynamic-options*, the statement is prepared as true dynamic SQL or XQuery statement and the generated plan is explained from the query cache. This explain-output method provides more accurate access plans than the **dynexpln** command, which prepares the statement as a static SQL or XQuery statement. The *dynamic-options* also allow the use of features available only in dynamic SQL and XQuery statements, such as parameter markers.

Also, you can use the **-opids** parameter in the **db2expln** command to print the operator ID number to the left of the explained plan in the same manner as the **dynexpln** command. You can continue to use these operator IDs to match up the steps in the different representations of the access plan.

Resolution

Start using the **db2expln** command instead before the **dynexpln** command becomes discontinued.

Also, modify any scripts or application that use the **dynexpln** command and replace it with the **db2expln** command.

PRUNE LOGFILE command has been deprecated

The **PRUNE LOGFILE** command has been deprecated and might be removed in a future release.

Details

For databases with log retain enabled and archive logging is not enabled, you can issue the **PRUNE LOGFILE** command to remove log files from the active log path that are not needed for recovery purposes.

For databases with archive logging enabled, the database manager copies the log files to an archive location and removes them from the active log path as needed. You can use the **PRUNE HISTORY** command to remove log files from the archive location when they are no longer needed for recovery purposes.

Resolution

Although you can use operating system commands to remove unnecessary log files, determining when it is safe to remove specific log files is not a trivial task. Instead, consider enabling your database for archive logging by setting database configuration parameter **logarchmeth1** to **USEREXIT**, **DISK**, **TSM**, or **VENDOR** and use the **PRUNE HISTORY** command to remove log files from the archive location.

Some parameters of the **CREATE DATABASE** command have been deprecated

The **AUTOMATIC STORAGE** parameter, the **MANAGED BY SYSTEM** clause in the **USER TABLESPACE** parameter, and values for **COLLATE USING** parameter of the **CREATE DATABASE** command have been deprecated.

Details

The **AUTOMATIC STORAGE** parameter, which indicates whether a database is enabled for automatic storage, is now deprecated. By default, all databases are enabled for automatic storage. DB2 pureScale only supports databases with automatic storage enabled. The IBMSTOGROUP storage group is automatically created as the default storage group in the SYSSTOGROUPS catalog table. Use the ALTER STOGROUP statement to change the default storage group or manage storage groups.

When you create a database, you can specify the table spaces you want to create for that database using the **MANAGED BY SYSTEM** parameter. The system managed spaces (SMS) table space type is now deprecated for permanent table spaces. The recommended table space types for user table spaces are automatic storage or database managed spaces (DMS).

The UCA400_NO, UCA400_LSK, and UCA400_LTH keywords for the **COLLATE USING** parameter of the **CREATE DATABASE** command have been deprecated. The supported locale-sensitive UCA-based collations provide the same functionality and better performance.

Resolution

Stop using the **AUTOMATIC STORAGE** parameter with the **CREATE DATABASE** command before it becomes discontinued. Also, remove this parameter from any scripts or applications.

Use automatic storage or DMS table space types to create new user table spaces. For existing SMS user table spaces, start converting them to automatic storage or DMS table space types table spaces before SMS user table spaces become discontinued. For more details, see “SMS permanent table spaces have been deprecated” on page 142.

Use any of the supported locale-sensitive UCA-based collations to create new Unicode databases. For existing databases, you can convert to a supported locale-sensitive UCA-based collation by recreating your database. For more details, see “Collations based on the Unicode Collation Algorithm of the Unicode Standard version 4.0.0 have been deprecated” on page 141.

Some parameters for commands that manage instances have changed

The **db2icrt**, **db2iupdt**, **db2cluster_prepare** commands have new parameters, changes to existing parameters, and deprecation of parameter value format.

Details

In Version 10.1, the **-m** and **-cf** parameters in the **db2icrt**, **db2iupdt**, and **db2cluster_prepare** commands indicate the hostname of a member or a cluster caching facility (CF) respectively. The cluster interconnect netname is the hostname

of the interconnect used for high speed communication between members and cluster caching facilities. To specify the cluster interconnect netname for members, use the new **-mnet** parameter. To specify the cluster interconnect netname for CFs, use the new **-cfnet** parameter.

In previous releases, you could use the **-m** and **-cf** parameters to indicate the hostname and the network name using a colon as separator. This format for these parameters is deprecated and might be discontinued in a future release.

If you want to indicate an IPv6 IP address for the hostname, you must indicate the hostname with the **-m** and **-cf** parameters, and the network name with the **-mnet** and **-cfnet** parameters. Using the old format available in previous releases to indicate both hostname and network name will return an error.

In addition, you can use the **instance_shared_mount** parameter in Version 10.1 to indicate the directory where you want to mount a newly created DB2 managed clustered file system. If you do not use this parameter, the database manager generates a name.

Resolution

Start using the new **-mnet** and **-cfnet** parameters to indicate the network name before the old format becomes discontinued. Modify any existing scripts and applications to use the new parameters.

Start using the **instance_shared_mount** parameter to indicate a directory to mount the DB2 managed clustered file system instead of having a system-generated name.

-flushbp parameter of the db2pdcfg command has been deprecated

The **-flushbp** parameter of the **db2pdcfg** command has been deprecated and might be removed in a future release.

Details

You must use the **-flushbp** of the **db2pdcfg** command only when instructed by IBM service.

In Version 10.1, the FLUSH BUFFERPOOLS statement writes the dirty pages from all the local buffer pools for a particular database synchronously to disk.

You can use the FLUSH BUFFERPOOLS statement to reduce the recovery window of a database in the event of a failure, or before database operations such as online backups. Also, to minimize the recovery time of the split-mirror database.

Resolution

Use the FLUSH BUFFERPOOLS statement instead.

REORG INDEXES/TABLE command parameters and parameter values for related DB2 API data structures have been deprecated or discontinued

To improve readability of the **REORG INDEXES/TABLE** command, certain command parameters have been deprecated and replaced by a new option. Also, some parameter values for related DB2 API data structures have been deprecated or discontinued.

Details

The **CLEANUP ONLY** parameter of the **REORG INDEXES** command has been deprecated and has been replaced by the **CLEANUP** option. The following example uses the **CLEANUP** option:

```
REORG INDEXES ALL FOR TABLE USER1.TABLE2 ALLOW WRITE ACCESS CLEANUP ALL;
```

The **RECLAIM EXTENTS ONLY** parameter of the **REORG TABLE** command has been deprecated and has been replaced by the **RECLAIM EXTENTS** option. The following example uses the **RECLAIM EXTENTS** option:

```
REORG TABLE USER1.TABLE1 RECLAIM EXTENTS;
```

The **CONVERT** parameter of the **REORG INDEXES** command has been discontinued. The type-1 indexes were discontinued in Version 9.7. Therefore, this parameter has become obsolete. Databases upgraded from Version 9.7 to Version 10.1 no longer have type-1 indexes. Databases upgraded from Version 9.5 to Version 10.1 have type-1 indexes automatically rebuilt as type-2 indexes on the first access to the table after the upgrade.

In the `db2ReorgStruct` data structure for the `db2Reorg` API, the following values for the `reorgFlags` parameter have been deprecated or discontinued:

- The `DB2REORG_CLEANUP_NONE` value has been deprecated. This value indicates that no clean up is required when the `reorgType` is set to `DB2REORG_OBJ_INDEXESALL` or `DB2REORG_OBJ_INDEX`. Not specifying this value has the same effect, therefore, specifying the value is redundant.
- The `DB2REORG_CONVERT_NONE` value has been deprecated. In earlier releases, this value indicated that no index conversion was required when the `reorgType` is set to `DB2REORG_OBJ_INDEXESALL` or `DB2REORG_OBJ_INDEX`. This value has become obsolete because type-1 indexes were discontinued since Version 9.7.
- The `DB2REORG_CONVERT` value has been discontinued. In earlier releases, this value indicated that index conversion was required when the `reorgType` is set to `DB2REORG_OBJ_INDEXESALL` or `DB2REORG_OBJ_INDEX`. This value has become obsolete because type-1 indexes were discontinued since Version 9.7.

In the `db2LoadQueryOutputStruct` data structure and `db2LoadQueryOutputStruct64` data structure for the `db2LoadQuery` API, the `DB2LOADQUERY_TYPE1_INDEXES` value for the `TableState` parameter has been discontinued. This value indicates table currently uses type-1 indexes and has become obsolete because type-1 indexes were discontinued since Version 9.7.

Resolution

Start using the CLEANUP option with the **REORG INDEXES** command or the RECLAIM EXTENTS option with the **REORG TABLE** command. You are no longer required to indicate the DB2REORG_CLEANUP_NONE in the reorgFlags parameter.

Stop using the **CONVERT** parameter of the **REORG INDEXES** command. If you use the parameter, an error is returned.

ALTER DATABASE statement has been deprecated

The ALTER DATABASE statement has been deprecated and might be removed in a future release. The CREATE STOGROUP or ALTER STOGROUP statements provide the same functionality as the ALTER DATABASE statement and more.

Details

In Version 10.1, you can issue the ALTER STOGROUP statement to add or remove storage paths to any storage group. In addition, you can use this statement to change the definition and attributes of a storage group. Use the CREATE STOGROUP statement to create a new storage group and assign storage paths to it.

With the ALTER DATABASE statement, you can only add or remove storage paths to the default storage group for the database. You cannot indicate a specific storage group.

Resolution

Start using the CREATE STOGROUP or ALTER STOGROUP statements to manage storage groups before the ALTER DATABASE statement becomes discontinued. Modify any scripts or application using the ALTER DATABASE statement and replace it with the CREATE STOGROUP or ALTER STOGROUP statements.

Take advantage of the new capabilities provided by the CREATE STOGROUP or ALTER STOGROUP statements to create new storage groups, set the default storage group, and change attributes of existing storage groups to improve the physical design of your database data storage.

Some registry and environment variables have been deprecated

There are a number of registry variables that are deprecated in Version 10.1. 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 33. Registry and environment variables deprecated in Version 10.1

Registry or environment variable	Details
DB2_LIKE_VARCHAR	The most recommended setting for this variable is Y, which is the default value. Configuring this variable on your own might result in unpredictable query performance because the scenarios that can benefit from setting the variable are complex. You should only change this variable at the recommendation of IBM Service.

Resolution

Stop using this registry variable unless you are directed to do so by IBM Service.

Some database configuration parameters have been deprecated or discontinued

Due to changes in functionality, the introduction of new database configuration parameters, or the removal of support, the following database configuration parameters have been either deprecated or discontinued.

Details

The following database configuration parameters are deprecated:

Table 34. Deprecated database configuration parameters

Parameter name	Description	Details
health_mon	Health monitoring configuration parameter	The health monitor and associated health indicators have been deprecated.
mincommit	Number of commits to group configuration parameter	Tuning of this parameter is not required due to improvements to the logging infrastructure.

The following database configuration parameters are discontinued:

Table 35. Discontinued database configuration parameters

Parameter name	Description	Details
dyn_query_mgmt	Dynamic SQL and XQuery query management	This parameter is no longer available because Query Patroller is discontinued.
logretain	Log retain enable	This parameter has been replaced with logarchmeth1 , the primary log archive method configuration parameter. To retain active log files for rollforward recovery, set logarchmeth1 to LOGRETAIN by issuing the command UPDATE DB CFG USING logarchmeth1 LOGRETAIN . logarchmeth2 must be set to OFF.

Table 35. Discontinued database configuration parameters (continued)

Parameter name	Description	Details
userexit	User exit enable	This parameter has been replaced with logarchmeth1 , the primary log archive method configuration parameter. To enable log archiving through a user exit program, set logarchmeth1 to USEREXIT by issuing the command UPDATE DB CFG USING logarchmeth1 USEREXIT . logarchmeth2 must be set to OFF.

The **logarchmeth1** and **logarchmeth2** configuration parameters in databases that were upgraded from pre-Version 10.1 releases have the same values as the **logretain** and **userexit** configuration parameters. In previous releases, the values of the **logarchmeth1** and **logarchmeth2** configuration parameters were synchronized to the values of the **logretain** and **userexit** configuration parameters. After upgrading your database, no action is required as the **logarchmeth1** and **logarchmeth2** configuration parameters are already set to the appropriate values.

Resolution

Remove the use of database configuration parameters that are deprecated because the functionality associated with the configuration parameter is obsolete or has been replaced by new functionality. Also, remove the use of discontinued database configuration parameters as they do not have the intended effect.

If a replacement database configuration parameter is provided, set it to maintain wanted database behavior.

Some monitoring interfaces for HADR have been deprecated

The SNAPHADR administrative view, SNAP_GET_HADR table function, and some HADR-related monitor elements have been deprecated in Version 10.1 and might be removed in a future release.

In Version 10.1, the MON_GET_HADR table function has been introduced to report information about HADR functionality. Also, the **db2pd -hadr** command output has been changed to report new HADR information. The HADR information reported by these interfaces include details about new functionality such as the HADR multiple standby mode.

You can still use the SNAPHADR administrative view and SNAP_GET_HADR table function, and the listed HADR-related monitor elements, but the HADR information reported does not include all of the details about new functionality. For example, if you call the SNAPHADR administrative view and SNAP_GET_HADR table function from the primary, they do not report information about the auxiliary standbys.

The following monitor elements have also been deprecated:

hadr_heartbeat - HADR heartbeat

To determine the equivalent information, use the **db2pd -hadr** command or the MON_GET_HADR table function to query the value for TIME_SINCE_LAST_RECV and divide it by HEARTBEAT_INTERVAL.

hadr_local_service - HADR local service

To determine the equivalent information, query the **hadr_local_svc** configuration parameter. On the HADR primary database, this refers to the primary, and on an HADR standby database, this refers to the standby on which the **DB2 GET CFG** command is issued.

hadr_remote_service - HADR remote service

To determine the equivalent information, query the **hadr_local_svc** and **hadr_remote_svc** configuration parameters. On the primary, this refers to the principal standby, and on a standby, this refers to primary database..

Resolution

Start using the **MON_GET_HADR** table function or the **db2pd -hadr** command to monitor HADR databases before the deprecated administrative view, table function, and monitor elements are discontinued.

The -global option for troubleshooting tools has been deprecated

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.

Chapter 22. Discontinued functionality

Discontinued functionality is removed in Version 10.1 and is no longer available. You need to make changes if you were using that functionality in previous releases.

DB2 functionality

The following DB2 functionality is discontinued:

- 32-bit client support (see “32-bit client support has been discontinued (HP-UX)” on page 156)
- DB2 JDBC Type 2 driver (see “DB2 JDBC Type 2 Driver has been discontinued” on page 156)
- DB2SE_USA_GEOCODER (see “DB2SE_USA_GEOCODER has been discontinued” on page 157)
- Distributed installation support with Microsoft Systems Management Server (see “Distributed installation support with Microsoft Systems Management Server has been discontinued (Windows)” on page 157)
- Some operating systems (see “Some operating systems are no longer supported” on page 158)
- Some version-suffixed SQL administrative routines (see “Some version-suffixed SQL administrative routines have been discontinued” on page 158)
- The Worksheet Format (WSF) for input or output files in the **IMPORT** and **EXPORT** commands (see “Worksheet Format (WSF) for Import and Export utilities have been discontinued” on page 159)
- Visual Studio 2005 support (see “Microsoft Visual Studio 2005 support has been discontinued” on page 159)

DB2 database products or features

The following products and features are discontinued:

- Query Patroller (see “Query Patroller has been discontinued” on page 159)
- Control Center tools (see “Control Center tools have been discontinued” on page 160)
- IBM DB2 Geodetic Data Management Feature (see “IBM DB2 Geodetic Data Management Feature has been discontinued” on page 163)

APIs, commands, command parameters, registry variables, and configuration parameters

The following APIs, commands, command parameters, and registry variable are discontinued:

- Support in COBOL, FORTRAN, and REXX for DB2 APIs that manage database history records has been discontinued (see “Support in COBOL, FORTRAN, and REXX for DB2 APIs that manage database history records has been discontinued” on page 163)
- The **db2imigr** and **db2ckmig** commands (see “db2imigr and db2ckmig commands have been discontinued” on page 164)
- The **-file** parameter of the **db2flsn** and **db2rfpen** commands (see “-file parameter of the db2flsn and db2rfpen commands has been discontinued” on page 164)

- The **-s** parameter of the **db2iupdt** command (see “-s parameter of the db2iupdt command has been discontinued” on page 165)
- The **CONVERT** parameter of the **REORG INDEXES** command and parameter values for related DB2 API data structures (see “REORG INDEXES/TABLE command parameters and parameter values for related DB2 API data structures have been deprecated or discontinued” on page 150)
- Some registry and environment variables (see “Some registry and environment variables have been discontinued” on page 165)
- Some database configuration parameters (see “Some database configuration parameters have been deprecated or discontinued” on page 152)

32-bit client support has been discontinued (HP-UX)

Starting with DB2 Version 10.1, 32-bit client support has been discontinued on HP-UX operating systems

Details

When support for HP Itanium-based servers was introduced in DB2 Version 8, 32-bit DB2 client libraries were provided on HP-UX operating systems 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 systems was removed in DB2 Version 9.5. As HP Itanium-based servers are now common, DB2 32-bit client support on HP-UX operating systems is discontinued. This change does not affect other platforms, where 32-bit DB2 client support continues to be enhanced.

Resolution

Support for 32-bit applications on HP-UX operating systems has been discontinued. You must upgrade your 32-bit applications to 64-bit before upgrading to Version 10.1, so that applications can run on native 64-bit HP-UX Itanium-based environments.

DB2 JDBC Type 2 Driver has been discontinued

The DB2 JDBC Type 2 driver for Linux, UNIX, and Windows operating systems has been discontinued. You should use the IBM Data Server Driver for JDBC and SQLJ instead.

Details

The DB2 JDBC Type 2 Driver for Linux, UNIX, and Windows was deprecated in Version 8.2. This driver allowed Java applications to make calls to DB2 servers through JDBC. The DB2 JDBC Type 2 Driver used the DB2 CLI interface to communicate with DB2 data servers. The Java applications that used this driver had to run on a DB2 client. This driver supported all of the methods that are described in the JDBC 1.2 and 2.0 specifications.

The IBM Data Server Driver for JDBC and SQLJ allows Java applications to make JDBC type 2 and JDBC type 4 connections using a single driver instance. The type 2 driver behavior of the IBM Data Server Driver for JDBC and SQLJ is referred to as IBM Data Server Driver for JDBC and SQLJ type 2 connectivity. This driver

supports all of the methods that are described in the JDBC 3.0 and 4.0 specifications.

Resolution

Use the IBM Data Server Driver for JDBC and SQLJ instead of the DB2 JDBC Type 2 Driver. Refer to the task Upgrading database applications at <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.qb.upgrade.doc/doc/t0024117.html> and subtask Upgrading Java applications that use DB2 JDBC Type 2 driver at <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.qb.upgrade.doc/doc/t0024117.html>.

If you use the DB2 JDBC Type 2 Driver to compile your Java applications or routines, you will receive an error as the driver cannot be found.

DB2SE_USA_GEOCODER has been discontinued

DB2SE_USA_GEOCODER is no longer supported. Also, it is no longer installed with DB2 Spatial Extender.

Details

The DB2SE_USA_GEOCODER is a sample geocoder that translates United States addresses into ST_Point data. In previous releases, it was installed as a component of the DB2 Spatial Extender and automatically registered. The geocoder was supported only on 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. Providing an update to that reference data would result in 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

Use 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*.

Distributed installation support with Microsoft Systems Management Server has been discontinued (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 discontinued.

Details

In Version 10.1, you can perform distributed installations of DB2 products using Microsoft Systems Center Configuration Manager (SCCM).

In previous releases, you could perform distributed installations using Microsoft Systems Management Server. However, Microsoft has ended support of SMS.

Resolution

Use SCCM to perform distributed installations instead. For details, see the SCCM documentation at <http://technet.microsoft.com/en-us/library/cc507089.aspx>.

Some operating systems are no longer supported

Starting in Version 10.1, support for some operating systems has been discontinued.

Details

The following operating systems and Linux distribution are no longer supported:

- AIX 5.3
- HP-UX 11iv2
- Solaris 9
- Ubuntu 8.0.4.x

User Response

Review the list of supported operating systems and plan to upgrade your DB2 server operating systems before upgrading your DB2 database products.

Some version-suffixed SQL administrative routines have been discontinued

A number of SQL administrative routines have been discontinued in Version 10.1. You should remove all references to these routines.

Details

In Version 9.1 and Version 9.5, some version-suffixed SQL administrative routines were deprecated. These routines have now been discontinued. The replacement routines in Version 10.1 have more comprehensive functionality.

The following table shows the routines that are now discontinued and their replacement routine:

Table 36. Discontinued SQL administrative routines and their replacement routines

Discontinued routine	Replacement routine
SNAP_GET_DB_V91	SNAP_GET_DB table function
SNAP_GET_DB_V95	SNAP_GET_DB table function
SNAP_GET_DYN_SQL_V91	SNAP_GET_DYN_SQL table function
SNAP_GET_TBSP_PART_V91	SNAP_GET_TBSP_PART table function

Note: If you call any of the discontinued routines, the SQL0440N error message is returned.

Resolution

Modify all your applications and scripts and remove all references to these routines or use the replacement routines or views indicated in “Deprecated SQL administrative routines and views”.

Worksheet Format (WSF) for Import and Export utilities have been discontinued

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 discontinued.

Details

WSF files have limitations relative to other supported file formats. This format is not recommended for DB2 utilities.

User Response

Use a supported file format instead of WSF files. Update any applications or scripts that use this file format for import and export operations. 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.

Microsoft Visual Studio 2005 support has been discontinued

Microsoft Visual Studio 2005 support has been discontinued because Visual Studio 2005 has reached its end of support date.

Details

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 through 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. If you perform additional work in Visual Studio 2005, you can use .NET Framework 3.0 or later because Visual Studio 2005 does not have any built-in dependency on the .NET Framework. However, you will not have access to new project and file templates and tools.

Resolution

Use Visual Studio 2008 or Visual Studio 2010 instead of Visual Studio 2005. You can convert existing projects in Visual Studio 2005 to Visual Studio 2008 or Visual Studio 2010.

Query Patroller has been discontinued

Query Patroller is discontinued and you must use the DB2 workload manager instead as your workload management solution.

Details

Query Patroller is no longer supported in Version 10.1. Query Patroller was deprecated in Version 9.7. As a result, related registry variables have also been discontinued.

In addition, the following keywords for the response file are discontinued:

- QUERY_PATROLLER_DATABASE
- QP_CONTROL_TABLESPACE
- QP_CONTROL_DBPARTITIONGROUP
- QP_CONTROL_DBPARTITIONNUM
- QP_CONTROL_PATH, QP_CONTROL_DMS
- QP_CONTROL_DMS_CONTAINER
- QP_CONTROL_DMS_NUMPAGES
- QP_RESULT_TABLESPACE
- QP_RESULT_DBPARTITIONGROUP
- QP_RESULT_DBPARTITIONNUM
- QP_RESULT_PATH, QP_RESULT_DMS
- QP_RESULT_DMS_CONTAINER
- QP_RESULT_DMS_NUMPAGES
- QP_REPLACE, QP_USERNAME
- QP_DOMAIN
- QP_PASSWORD

DB2 workload manager (WLM) is the preferred workload management solution since Version 9.5. It provides a greatly enhanced set of workload management functions that replace both Query Patroller and DB2 Governor.

Resolution

Starting in Version 9.7 Fix Pack 1, you can use a script called `qpwlmmig.pl` that generates a script that helps to migrate a Query Patroller environment to a WLM environment. See one of the following tasks for details on how to migrate from Query Patroller to DB2 workload manager:

- Migrating from Query Patroller to DB2 workload manager using the sample script
- Migrating from Query Patroller to DB2 workload manager

You must run the `qpwlmmig.pl` script in your Version 9.7 DB2 copy where Query Patroller is installed before upgrading to Version 10.1. Query Patroller is not supported in Version 10.1. The `qpwlmmig.pl` script also runs in Version 9.5 DB2 copies where Query Patroller is installed.

Control Center tools have been discontinued

The Control Center tools and all related components such as wizards and advisors are discontinued. A new suite of GUI tools for managing DB2 for Linux, UNIX, and Windows data and data-centric applications is now available and can be used instead.

Details

The following Control Center tools and related features have been discontinued:

- 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 wizard
 - 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 wizard
 - 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, there are commands and response file keywords for DB2 installation that are also discontinued. The following associated DB2 commands are discontinued:

- **db2am** (Start Activity monitor center command)
- **db2ca** (Start the Configuration Assistant command)
- **db2cc** (Start control center command)
- **db2ce** (Start Command Editor command)
- **db2eva** (Event analyzer command)
- **db2hc** (Start health center command)
- **db2indbt** (Start Indoubt Transaction Monitor Center command)
- **db2journal** (Start Journal command)
- **db2lc** (Start License Center command)
- **db2memvis** (Start Memory Visualizer center command)
- **db2tc** (Start Task Center command)

In addition, the following response file keywords are discontinued:

- CTLSRV_INSTANCE
- DB2SATELLITEAPPVER
- DB2SATELLITEID
- SATELLITE_CONTROL_DATABASE
- TOOLS_CATALOG_DATABASE
- TOOLS_CATALOG_SCHEMA

For Query Patroller, the response file keywords that are discontinued are listed at “Query Patroller has been discontinued” on page 159.

Important: Use of the Control Center tools Version 9.7 or earlier to connect to Version 10.1 databases is not supported. If you set the location to access the Version 10.1 DB2 Information Center by changing the DB2_DOCHOST variable, the online help for Control Center tools Version 9.7 or earlier is not available.

Resolution

Use the **AUTOCONFIGURE** command to get recommendations from the configuration advisor. Although the wizard interface for the configuration advisor is discontinued, the configuration advisor is still available by using the **AUTOCONFIGURE** command.

Use the **db2adv** command to get recommendations from the design advisor. The wizard interface for the design advisor is also discontinued, but the design advisor is still available using the **db2adv** command.

IBM Data Studio and IBM Optim tools perform similar tasks that you used to perform with the Control Center tools. These recommended tools provide enhanced features for those tasks. Use any of the following tools in place of the Control Center tools:

- IBM Data Studio
- IBM InfoSphere Data Architect
- IBM InfoSphere Optim Database Administrator
- IBM InfoSphere Optim Development Studio
- IBM InfoSphere Optim Performance Manager
- IBM InfoSphere Optim Performance Manager Extended Edition

For a mapping between these recommended tools and Control Center tools, see “Table of recommended tools versus Control Center tools” in the *What’s New for DB2 Version 10.1* book.

Data Studio, IBM InfoSphere Optim Database Administrator, and IBM InfoSphere Optim Development Studio are included in all the DB2 database product editions. IBM InfoSphere Optim Performance Manager is included with the DB2 Advanced Enterprise Server Edition or IBM InfoSphere Warehouse.

For more information, see “Database management and application development tools”.

Related information:

 Integrated Data Management Information Center at <http://publib.boulder.ibm.com/infocenter/idm/docv3/index.jsp>

IBM DB2 Geodetic Data Management Feature has been discontinued

The DB2 Geodetic Data Management Feature has been discontinued. If you hold a license for this feature from earlier releases, you will continue to receive support.

Details

In Version 10.1, the DB2 Spatial Extender installation does not include the DB2 Geodetic Data Management Feature. For new installations of the DB2 Spatial Extender with the DB2 Geodetic Data Management Feature, contact IBM support for details on how to obtain the software for this feature.

In 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.

This feature was deprecated in Version 9.7.

Resolution

If you hold a license for the DB2 Geodetic Data Management Feature in Version 9.7 or Version 9.5 and you upgrade DB2 Spatial Extender to Version 10.1, follow the instructions provided by IBM support to download and install DB2 Geodetic Data Management Feature and feature license.

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.

Support in COBOL, FORTRAN, and REXX for DB2 APIs that manage database history records has been discontinued

The support in COBOL, FORTRAN, and REXX for DB2 APIs that manage database history records has been discontinued. In Version 10.1, these APIs are still supported for C and Java programming languages.

Details

The support in COBOL, FORTRAN, and REXX has been removed for the following DB2 APIs:

- db2HistoryCloseScan
- db2HistoryGetEntry
- db2HistoryOpenScan
- db2HistoryUpdate

Resolution

Modify all your applications programmed in COBOL, FORTRAN, and REXX and scripts and remove all references to these DB2 APIs.

To access database history records, you can issue a query using the DB_HISTORY administrative view.

db2imigr and db2ckmig commands have been discontinued

The **db2imigr**, and **db2ckmig** commands have been discontinued.

Details

In Version 9.5 and earlier releases, the term *migration* was used to describe the process of upgrading DB2 servers, clients, database applications, routines, instances, and databases from a previous release to run in an environment of the current release. For consistency across DB2 products terminology, starting with Version 9.7, the term *upgrade* began to be used to describe the same process.

Because of this change in terminology, the DB2 commands to migrate instances and databases have been discontinued. The following table shows the commands to use for upgrading instances and databases.

Table 37. Discontinued and replacement commands

Discontinued command	Replacement command	Replacement command description
db2imigr	db2iupgrade	The db2iupgrade command upgrades an instance from a DB2 copy of a previous release to a DB2 copy of the current release.
db2ckmig	db2ckupgrade	db2ckupgrade verifies that local databases are ready to be upgraded.

User response

Use the **db2iupgrade** and **db2ckupgrade** commands when upgrading instances and databases.

-file parameter of the db2flsn and db2rfpen commands has been discontinued

The **-file** parameter of the **db2flsn** and **db2rfpen** commands has been discontinued. You must use the **-path** parameter instead.

Details

The **-file** parameter was discontinued because you could specify only one log control file (SQLOGCTL.LFH.1 or SQLOGCTL.LFH.2) with the parameter. For the **db2rfsn** command, specifying only one log control file can result in an inconsistency between the two log control files which can lead to potential data consistency issues in the database. For the **db2f1sn** command, if there is a problem with the file specified then an error is returned.

The **-path** parameter specifies the full path to the directory where the log control files, SQLOGCTL.LFH.1 and its mirror copy SQLOGCTL.LFH.2, reside. Using a path for the **db2rfsn** command means that both log control files in the path are updated. Using a path for the **db2f1sn** command means that if there is a problem reading one of the files, the other one is used instead.

Resolution

Use the **database_alias** parameter or the **-path** parameter instead.

-s parameter of the db2iupdt command has been discontinued

The **-s** parameter of the **db2iupdt** command is no longer available because the existing SPM log directory can not be ignored.

Details

The **db2iupdt** command performs the following functions:

- Updates an instance to run on a DB2 copy that has a new DB2 database product or feature installed
- Updates an instance to run on a DB2 copy of the same version as the DB2 copy that is associated with the instance
- Updates an instance type to a later-release instance type

In previous releases on UNIX and Linux operating systems, you could specify the **-s** parameter to ignore the existing sync point manager (SPM) log directory.

User response

On all operating systems, this parameter of the **db2iupdt** command is no longer available. On UNIX and Linux operating systems, do not indicate this parameter with the **db2iupdt** command. If you indicate this parameter, the existing sync point manager (SPM) log directory will not be ignored and an error will be returned.

Some registry and environment variables have been discontinued

There are a number of registry variables that are discontinued in Version 10.1. You should remove all references to them.

Details

The following registry and environment variables are discontinued in Version 10.1:

Table 38. Registry variables discontinued in Version 10.1

Registry or environment variable	Details
DB2_ASYNC_IO_MAXFILOP	This variable is obsolete because of the shared file handle table maintained by the threaded database manager.
DB2_BAR_AUTONOMIC_DISABLE	This variable was needed for IBM internal use only.
DB2COUNTRY	This variable is replaced with the DB2TERRITORY registry variable. Use the DB2TERRITORY registry variable to specify the region or territory code of a client application, which influences the date and time formats. DB2TERRITORY accepts the same values as DB2COUNTRY : for example, setting DB2COUNTRY to 68 is equivalent to setting DB2TERRITORY to 68.
DB2DEFPREP	This variable was needed only when using old versions of DB2 where the DEFERRED_PREPARE precompile parameter was not available.
DB2_DJ_COMM	This variable was used to specify the wrapper libraries that are loaded when the database manager is started. The wrapper library structure and loading method have since been enhanced making this variable obsolete.
DB2DMNBCKCTLR	This variable is no longer necessary because back up domain controllers in the Active Directory are only on the Windows NT platform, not on the Windows 2003 and Windows XP Professional platforms. Newer versions of DB2 do not run on the Windows NT platform.
DB2FFDC	This variable is replaced with the DB2FODC registry variable. The same functionality that DB2FFDC provided is available if you use the DUMPCORE parameter of DB2FODC . By default, the DUMPCORE parameter is set to ON to enable core file generation and to maintain compatibility with previous releases.
DB2_HASH_JOIN	This variable, created to provide control over the join method called hash join, is no longer necessary. The query optimizer automatically determines the best join method including hash join.
DB2_MAP_XML_AS_CLOB_FOR_DLC	This variable is discontinued because most existing DB2 applications that access XML values do so with an XML-capable client (Version 9.1 and newer). You need this variable only for previous applications that generically fetched table data and could not parse UTF-8 XML data in a BLOB.
DB2MEMMAXFREE	This variable is no longer necessary because the database manager now uses a threaded engine model. For more information, see The DB2 Process Model.
DB2_QP_BYPASS_APPLICATIONS	This variable is no longer supported because the functionality provided by the DB2 Query Patroller has been replaced by the DB2 workload manager.
DB2_QP_BYPASS_COST	This variable is no longer supported because DB2 Query Patroller is discontinued. The DB2 workload manager feature replaces DB2 Query Patroller and provides a complete solution.
DB2_QP_BYPASS_USERS	This variable is no longer supported because DB2 Query Patroller is discontinued. The DB2 workload manager feature replaces DB2 Query Patroller and provides a complete solution.
DB2ROUTINE_DEBUG	This variable is no longer necessary because this stored procedure debugger has been replaced by the unified debugger.

Table 38. Registry variables discontinued in Version 10.1 (continued)

Registry or environment variable	Details
DB2_RR_TO_RS	This variable is discontinued because Type-1 indexes are no longer supported.
DB2_SNAPSHOT_NOAUTH	This variable is not needed because you can achieve the same functionality by using the SYSMON authority group.
DB2_UPDATE_PART_KEY	This variable is obsolete because partitioning key updates are permitted by default.
DB2_USE_DB2JCCT2_JROUTINE	This variable is no longer needed because the driver it relates to has been discontinued.
DB2_VENDOR_INI	This variable is no longer necessary because you can put the environment variable settings that it contains into the file specified by the DB2_DJ_INI variable.
DB2YIELD	This variable was only used on Windows 3.1, which is not supported by newer versions of DB2
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 	These variables are no longer supported because DB2 Query Patroller is discontinued. DB2 workload manager feature replaces DB2 Query Patroller and provides a more complete solution.

Resolution

Remove the use of registry variables that are discontinued as they do not have the intended effect. If a replacement registry variable is indicated in Table 38 on page 166, set it to the proper value to maintain wanted database manager behavior.

Some database configuration parameters have been deprecated or discontinued

Due to changes in functionality, the introduction of new database configuration parameters, or the removal of support, the following database configuration parameters have been either deprecated or discontinued.

Details

The following database configuration parameters are deprecated:

Table 39. Deprecated database configuration parameters

Parameter name	Description	Details
health_mon	Health monitoring configuration parameter	The health monitor and associated health indicators have been deprecated.
mincommit	Number of commits to group configuration parameter	Tuning of this parameter is not required due to improvements to the logging infrastructure.

The following database configuration parameters are discontinued:

Table 40. Discontinued database configuration parameters

Parameter name	Description	Details
dyn_query_mgmt	Dynamic SQL and XQuery query management	This parameter is no longer available because Query Patroller is discontinued.
logretain	Log retain enable	This parameter has been replaced with logarchmeth1 , the primary log archive method configuration parameter. To retain active log files for rollforward recovery, set logarchmeth1 to LOGRETAIN by issuing the command UPDATE DB CFG USING logarchmeth1 LOGRETAIN . logarchmeth2 must be set to OFF.
userexit	User exit enable	This parameter has been replaced with logarchmeth1 , the primary log archive method configuration parameter. To enable log archiving through a user exit program, set logarchmeth1 to USEREXIT by issuing the command UPDATE DB CFG USING logarchmeth1 USEREXIT . logarchmeth2 must be set to OFF.

The **logarchmeth1** and **logarchmeth2** configuration parameters in databases that were upgraded from pre-Version 10.1 releases have the same values as the **logretain** and **userexit** configuration parameters. In previous releases, the values of the **logarchmeth1** and **logarchmeth2** configuration parameters were synchronized to the values of the **logretain** and **userexit** configuration parameters. After upgrading your database, no action is required as the **logarchmeth1** and **logarchmeth2** configuration parameters are already set to the appropriate values.

Resolution

Remove the use of database configuration parameters that are deprecated because the functionality associated with the configuration parameter is obsolete or has been replaced by new functionality. Also, remove the use of discontinued database configuration parameters as they do not have the intended effect.

If a replacement database configuration parameter is provided, set it to maintain wanted database behavior.

Chapter 23. Summary of deprecated and discontinued DB2 functionality in Version 10.1 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 9.5, Version 9.7, Version 9.8, or Version 10.1.

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 9.5 or earlier releases and possibly discontinued in a later release” on page 172
- “Functionality deprecated in Version 9.7 and possibly discontinued in a later release” on page 180
- “Functionality deprecated in Version 10.1 and possibly discontinued in a later release” on page 186

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 by DB2 database product release, use the following information:

Table 41. Discontinued functionality by DB2 database product release

Release	Links to additional information
Version 9.5	<ul style="list-style-type: none">• See “Discontinued functionality summary” at http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.wn.doc/doc/c0023234.html• See “Some 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">• See “Discontinued functionality summary” at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/c0023234.html
Version 10.1	<ul style="list-style-type: none">• See “Discontinued functionality summary” at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/c0023234.html• See “Some 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

Functionality deprecated in Version 9.5 or earlier releases and possibly discontinued in a later release

Table 42. Functionality deprecated in Version 9.5 or earlier releases 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	DROP PARTITIONING KEY clause of the ALTER TABLE statement is deprecated. See "ADD PARTITIONING KEY clause of the ALTER TABLE statement is deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.rn.doc/doc/c0023217.htm and "DROP PARTITIONING KEY clause of the ALTER TABLE statement is deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.rn.doc/doc/c0023216.htm
agentpri configuration parameter	To be determined	See "Some database manager configuration parameters have changed" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.wn.doc/doc/i0052553.html
app_ct1_heap_sz , appgroup_mem_sz , and groupheap_ratio configuration parameters	To be determined	See "Some database configuration parameters have changed" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.wn.doc/doc/i0052508.html
COLNAMES column in SYSCAT.INDEXES	To be determined	See "COLNAMES column in SYSCAT.INDEXES is deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.rn.doc/doc/c0023225.htm
Database logging using raw devices	To be determined	See "Database logging using raw devices is deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.rn.doc/doc/c0023086.htm
db2ilist command options -a and -p (Linux and UNIX operating systems)	Version 9.7	See "db2ilist command options -a and -p have been discontinued" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0054579.html
db2secv82 command	Version 9.7	See "db2secv82 command has been discontinued" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0054247.html

Table 42. Functionality deprecated in Version 9.5 or earlier releases and possibly discontinued in a later release (continued)

Functionality	Discontinued in release	Links to additional information
Default function entry points support in external routine libraries	To be determined	See "External routines now require an explicit entry point specification" at http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.rn.doc/doc/c0023021.htm
DB2 embedded application server (EAS)	Version 9.7	See "DB2 embedded application server (EAS) has been discontinued" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0054935.html
GET AUTHORIZATIONS command	Version 9.7	See "GET AUTHORIZATIONS command has been discontinued" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0054248.html
iCheckPending parameter	To be determined	Check pending table state is replaced and iCheckPending parameter is deprecated. See "Check pending table state is replaced and iCheckPending parameter is deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.rn.doc/doc/c0024079.htm
IMPORT command options CREATE and REPLACE_CREATE	To be determined	See "IMPORT command options CREATE and REPLACE_CREATE are deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.wn.doc/doc/i0052280.html
logretain and userexit configuration parameter	Version 10.1	See "Some configuration parameters have been deprecated and discontinued" at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/i0058741.html
Log control file SQLLOGCTL.LFH and the -file option of the db2f1sn command	To be determined	See "Log control file SQLLOGCTL.LFH has been renamed and copied" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.wn.doc/doc/i0051995.html
LONG VARCHAR and LONG VARGRAPHIC data types	To be determined	See "LONG VARCHAR and LONG VARGRAPHIC data types have been deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.wn.doc/doc/i0053661.html

Table 42. Functionality deprecated in Version 9.5 or earlier releases and possibly discontinued in a later release (continued)

Functionality	Discontinued in release	Links to additional information
maxagents and maxcagents configuration parameters	To be determined	See "Some database manager configuration parameters have changed" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.wn.doc/doc/i0052553.html
Netscape browser support	Version 9.7	See "Netscape browser support has been discontinued" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0054105.html
Network Information Services (NIS and NIS+) support and related registry variable (Linux and UNIX operating systems)	To be determined	See "Network Information Services (NIS and NIS+) support is deprecated (Linux and UNIX)" at http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.rn.doc/doc/c0024980.htm
numsegs configuration parameter	To be determined	See "Some database configuration parameters have changed" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.wn.doc/doc/i0052508.html
query_heap_sz configuration parameter	To be determined	See "Some database manager configuration parameters have changed" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.wn.doc/doc/i0052553.html
sqluadau API	Version 9.7	See "sqluadau API and sql_authorization data structure have been discontinued" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0054249.html
Static data stream snapshot output	To be determined	See "Static data stream snapshot output is deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.wn.doc/doc/i0052585.html
SNAP_GET_DB_V91, SNAP_GET_DB_V95, and SNAP_GET_DYN_SQL_V91 table functions	Version 10.1	See "Some version-suffixed SQL administrative routines have been discontinued" at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/i0058675.html

Table 42. Functionality deprecated in Version 9.5 or earlier releases and possibly discontinued in a later release (continued)

Functionality	Discontinued in release	Links to additional information
Type-1 indexes and related functionality	Version 9.7	See "Type-1 indexes have been discontinued" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0054145.html
Web Object Runtime Framework (WORF) support	Version 9.7	See "Web Object Runtime Framework (WORF) support has been discontinued" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0054173.html
XML Extender	Version 9.7	See "XML Extender has been discontinued" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0052510.html
HP-UX 32-bit client support	To be determined	See "HP-UX 32-bit client support has been deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.wn.doc/doc/i0058505.html
DB2 Health Advisor	To be determined	See "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 43. 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	See "Some registry and environment variables are deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.wn.doc/doc/i0052033.html
DB2ATLD_PORTS	To be determined	See "Some registry and environment variables are deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.wn.doc/doc/i0052033.html

Table 43. 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_ASYNC_IO_MAXFILOP	Version 10.1	See "Some 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
DB2_BAR_AUTONOMIC_DISABLE	Version 10.1	See "Some 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
DB2BPVARS	To be determined	See "Some registry and environment variables are deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.wn.doc/doc/i0052033.html
DB2_COMMIT_ON_EXIT	To be determined	See the "Deprecated registry variables in Version 9.1 table" at http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=/com.ibm.db2.udb.admin.doc/doc/r0004670.htm%23r0004670__depr-rv
DB2COUNTRY	Version 10.1	See "Some 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
DB2_CORRELATED_PREDICATES	To be determined	See the "Deprecated registry variables in Version 9.1 table" at http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=/com.ibm.db2.udb.admin.doc/doc/r0004670.htm%23r0004670__depr-rv
DB2DEFPREP	Version 10.1	See "Some 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
DB2_DJ_COMM	Version 10.1	See "Some 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

Table 43. Registry variables deprecated in Version 9.5 and possibly discontinued in a later release (continued)

Registry variable	Discontinued in release	Links to additional information
DB2DMNBCKCTLR	Version 10.1	See "Some registry and environment variables are discontinued" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.wn.doc/doc/i0052033.html
DB2_ENABLE_BUFDP	To be determined	See the "Deprecated registry variables in Version 9.1 table " at http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=/com.ibm.db2.udb.admin.doc/doc/r0004670.htm%23r0004670__depr-rv
DB2_ENABLE_SINGLE_NIS_GROUP	To be determined	See the "Deprecated registry variables in Version 9.1 table " at http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=/com.ibm.db2.udb.admin.doc/doc/r0004670.htm%23r0004670__depr-rv
DB2FFDC	Version 10.1	See "Some 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
DB2_HASH_JOIN	Version 10.1	See "Some 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
DB2_INDEX_FREE	To be determined	See "Some registry and environment variables are deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.wn.doc/doc/i0052033.html
DB2_MAP_XML_AS_CLOB_FOR_DLC	Version 10.1	See "Some 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
DB2_MAPPED_BASE	To be determined	See the "Deprecated registry variables in Version 9.1 table " at http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=/com.ibm.db2.udb.admin.doc/doc/r0004670.htm%23r0004670__depr-rv

Table 43. Registry variables deprecated in Version 9.5 and possibly discontinued in a later release (continued)

Registry variable	Discontinued in release	Links to additional information
DB2MEMMAXFREE	Version 10.1	See "Some 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
DB2_NO_MPFA_FOR_NEW_DB	To be determined	See the "Deprecated registry variables in Version 9.1 table" at http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=/com.ibm.db2.udb.admin.doc/doc/r0004670.htm%23r0004670__depr-rv
DB2_PARTITIONEDLOAD_DEFAULT	To be determined	See "Some registry and environment variables are deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.wn.doc/doc/i0052033.html
DB2_PRED_FACTORIZE	To be determined	See the "Deprecated registry variables in Version 9.1 table " at http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=/com.ibm.db2.udb.admin.doc/doc/r0004670.htm%23r0004670__depr-rv
DB2PRIORITIES and DB2NTPRICLASS	To be determined	See "Some registry and environment variables are deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.wn.doc/doc/i0052033.html
DB2ROUTINE_DEBUG	Version 10.1	See "Some 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
DB2_RR_TO_RS	Version 10.1	See "Some 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
DB2_SNAPSHOT_NOAUTH	Version 10.1	See "Some 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

Table 43. 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_TRUSTED_BINDIN	To be determined	See "Some registry and environment variables are deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.wn.doc/doc/i0052033.html
DB2_UPDATE_PART_KEY	Version 10.1	See "Some 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
DB2_VENDOR_INI	Version 10.1	See "Some 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
DB2YIELD	Version 10.1	See "Some 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

Table 43. Registry variables deprecated in Version 9.5 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 	Version 10.1	See "Some 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

Functionality deprecated in Version 9.7 and possibly discontinued in a later release

Table 44. Functionality deprecated in Version 9.7

Functionality	Discontinued in release	Links to additional information
CREATE EVENT MONITOR FOR DEADLOCKS statement and DB2DETAILDEADLOCK event monitor	To be determined	See "CREATE EVENT MONITOR FOR DEADLOCKS statement and DB2DETAILDEADLOCK event monitor have been deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0054715.html

Table 44. Functionality deprecated in Version 9.7 (continued)

Functionality	Discontinued in release	Links to additional information
CREATE EVENT MONITOR FOR TRANSACTIONS statement	To be determined	See "CREATE EVENT MONITOR FOR TRANSACTIONS statement has been deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0054716.html
DB2 Governor	To be determined	See "DB2 Governor and Query Patroller have been deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0054901.html
Query Patroller	Version 10.1	See "Query Patroller has been discontinued" at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/i0058730.html
IBM DB2 Geodetic Data Management Feature	Version 10.1	See "IBM DB2 Geodetic Data Management Feature has been discontinued" at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/i0059833.html
db2imigr and db2ckmig commands	Version 10.1	See "db2imigr and db2ckmig commands have been discontinued" at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/i0058593.html
MIGRATE DATABASE commands; sqlmgdb and sqlmgdb APIs	To be determined	See "Instance and database migration commands and APIs have been deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0053793.html
-s parameter of the db2iupdt command	Version 10.1	See " -s parameter of the db2iupdt command has been discontinued" at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/i0058592.html
-file parameter of the db2rfsn and db2rfpen command	Version 10.1	See " -file parameter of the db2rfsn and db2rfpen commands has been discontinued" at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/i0058592.html

Table 44. Functionality deprecated in Version 9.7 (continued)

Functionality	Discontinued in release	Links to additional information
Worksheet Format (WSF) for Import and Export utilities	Version 10.1	See "Worksheet Format (WSF) for Import and Export utilities have been discontinued" at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/i0057410.html
LIST TABLESPACES and LIST TABLESPACE CONTAINERS commands and related APIs: <ul style="list-style-type: none"> • sqlbctsq • sqlbftsq • sqlbftpq • sqlbgtss • sqlbmtsq • sqlbotsq • sqlbstpq • sqlbstsq • sqlbtcq 	To be determined	See "LIST TABLESPACES and LIST TABLESPACE CONTAINERS commands have been deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0055001.html
MIGRATE_PRIOR_VERSIONS and the CONFIG_ONLY response file keywords	To be determined	See "Some response file keywords have been deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0054093.html
CONVERT parameter of the REORG INDEXES command	Version 10.1	See "Type-1 indexes have been discontinued" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0054145.html
DB2LOADQUERY_TYPE1_INDEXES and the DB2REORG_CONVERT values in certain DB2 API data structures	Version 10.1	See "Type-1 indexes have been discontinued" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0054145.html
sqlugtpi API	To be determined	See "sqlugtpi API has been deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0053643.html
sqlugrpn API	To be determined	See "sqlugrpn API has been deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0054733.html

Table 44. Functionality deprecated in Version 9.7 (continued)

Functionality	Discontinued in release	Links to additional information
db2HistoryCloseScan, db2HistoryGetEntry, db2HistoryOpenScan, and db2HistoryUpdate APIs	Version 10.1	See "Support in COBOL, FORTRAN, and REXX for DB2 APIs that manage database history records has been discontinued" at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/i0059298.html
DB2SE_USA_GEOCODER	Version 10.1	See "DB2SE_USA_GEOCODER has been discontinued" at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/i0059643.html
Subset of Net Search Extender features and commands	To be determined	See "Subset of Net Search Extender features and commands have been deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0054683.html
Subset of SQL administrative routines	To be determined	See "Deprecated Version 9.7 SQL administrative routines and their replacement routines or views" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.sql.rtn.doc/doc/r0023171.html
SNAP_GET_TBSP_PART_V91 table function	Version 10.1	See "Some version-suffixed SQL administrative routines have been discontinued" at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/i0058675.html

Table 44. Functionality deprecated in Version 9.7 (continued)

Functionality	Discontinued in release	Links to additional information
<p>The following DB2 administration tools:</p> <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 	Version 10.1	<p>See “DB2 administration tools have been discontinued” at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/i0058731.html</p>
DB2 administration server (DAS)	To be determined	<p>See “Control Center tools and DB2 administration server (DAS) have been deprecated” at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0054250.html</p>
DB2 Health Advisor	To be determined	<p>See “DB2 Health Advisor has been deprecated” at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0059075.html</p>
Health monitor	To be determined	<p>See “Health monitor has been deprecated” at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0055045.html</p>
HP-UX 32-bit client support	To be determined	<p>See “HP-UX 32-bit client support has been deprecated” at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0058505.html</p>

Table 44. Functionality deprecated in Version 9.7 (continued)

Functionality	Discontinued in release	Links to additional information
Visual Studio 2005 support	Version 10.1	See "Visual Studio 2005 support has been discontinued" at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/i0060222.html

Table 45. Registry variables deprecated in Version 9.7

Registry variables	Discontinued in release	Links to additional information
DB2_CAPTURE_LOCKTIMEOUT	To be determined	See "Some registry and environment variables are deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0052033.html
DB2_QP_BYPASS_APPLICATIONS	Version 10.1	See "Some registry and environment variables are deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0052033.html
DB2_QP_BYPASS_COST	Version 10.1	See "Some registry and environment variables are deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0052033.html
DB2_QP_BYPASS_USERS	Version 10.1	See "Some registry and environment variables are deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0052033.html
DB2_SERVER_ENCALG	To be determined	See "Some registry and environment variables are deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0052033.html
DB2_USE_DB2JCCT2_JROUTINE	Version 10.1	See "Some registry and environment variables are deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.wn.doc/doc/i0052033.html

Functionality deprecated in Version 10.1 and possibly discontinued in a later release

Table 46. Functionality deprecated in Version 10.1

Functionality	Discontinued in release	Links to additional information
Activity monitor routines	To be determined	See "Activity monitor routines have been deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/i0059660.html
Agent priority of service classes	To be determined	See "Agent priority of service classes has been deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/i0059642.html
Collations based on the Unicode Standard version 4.0.0	To be determined	See "Collations based on the Unicode Standard version 4.0.0 have been deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/i0058749.html
INSTALL_TSAMP response file keyword	To be determined	See "IBM Tivoli System Automation for Multiplatforms (SA MP) is now automatically installed" at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/i0059123.html
mincommit database configuration parameter	To be determined	See "Some configuration parameters have been deprecated and discontinued" at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/i0058741.html
Net Search Extender	To be determined	See "Net Search Extender has been deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/i0058722.html
SMS user table spaces	To be determined	See "SMS user table spaces have been deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/i0058748.html
Automatic statistics profiling	To be determined	See "Automatic statistics profiling has been deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/i0060240.html

Table 46. Functionality deprecated in Version 10.1 (continued)

Functionality	Discontinued in release	Links to additional information
Subset of version-suffixed SQL administrative routines	To be determined	See "Some version-suffixed SQL administrative routines have been deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/i0058674.html
-flushbp parameter of the db2pdcfg command	To be determined	See " -flushbp parameter of the db2pdcfg command has been deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/i0058772.html
One column output by the -tcbstat parameter of the db2pd command	To be determined	See "One column output by the -tcbstat parameter of the db2pd command has been deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/i0058932.html
Some parameters of the CREATE DATABASE command	To be determined	See "Some parameters of the CREATE DATABASE command have been deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/i0058750.html
db2IdentifyType1 command	To be determined	See " db2IdentifyType1 command has been deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/i0059642.html
db2_install command	To be determined	See " db2_install command has been deprecated (Linux and UNIX)" at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/i0058736.html
dynexpln command	To be determined	See " dynexpln command has been deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/i0058770.html
PRUNE LOGFILE command	To be determined	See " PRUNE LOGFILE command has been deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/i0058771.html

Table 46. Functionality deprecated in Version 10.1 (continued)

Functionality	Discontinued in release	Links to additional information
Some options of the REORG INDEXES/TABLE command and parameter values for related DB2 API data structures	To be determined	See "REORG INDEXES/TABLE command parameters and parameter values for related DB2 API data structures have been deprecated or discontinued" at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/i0059689.html
Format of the -m and -cf parameters in some commands that manage instances	To be determined	See "Some commands that manage instances have changed" at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/i0059122.html
ALTER DATABASE statement	To be determined	See "ALTER DATABASE statement has been deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/i0058773.html
ENHANCED_MULTIPLE_DISTINCT setting for DB2_EXTENDED_OPTIMIZATION	To be determined	See "Some registry and environment variables have changed" at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/i0052034.html

Table 47. Registry variables and settings that are deprecated in Version 10.1

Registry variables	Discontinued in release	Links to additional information
DB2_LIKE_VARCHAR	To be determined	See "Some registry and environment variables are deprecated" at http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.db2.luw.wn.doc/doc/i0052033.html

Part 3. DB2 Connect enhancements and changes summary

In Version 10.1, DB2 Connect product capabilities have been enhanced and changed.

Chapter 24, “DB2 Version 10.1 enhancements and changes that affect DB2 Connect,” on page 191

This chapter describes the enhancements, changed functionality, deprecated functionality, and discontinued functionality in Version 10.1 that affect the functionality and capabilities of DB2 Connect.

Related concepts:

Part 2, “What's changed,” on page 97

Chapter 24. DB2 Version 10.1 enhancements and changes that affect DB2 Connect

Because of the common functionality shared between DB2 database products and DB2 Connect, some of the DB2 Version 10.1 enhancements and changes affect the functionality and capabilities of DB2 Connect.

The following Version 10.1 enhancements and changes affect the DB2 Connect functionality. 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.

Installation, upgrade, and fix pack enhancements

- New command parameters (see "Installation commands have been enhanced" on page 81)
- New response file keywords have been added to support new features (see "New response file keywords have been added" on page 83)
- License violations reported (see "DB2 License compliance report enhanced" on page 83)
- New command to check prerequisites before installation (see "New db2prereqcheck command checks prerequisites before starting an installation" on page 83)

Administration changes

- Replication Center is now a stand-alone tool (see "Replication Center is now a stand-alone tool" on page 99)
- db2ckupgrade command now verifies that all database partitions are ready for upgrade (see "Replication Center is now a stand-alone tool" on page 99)

Security changes

- Kerberos authentication support for several UNIX operating systems has changed (see "Kerberos authentication changes (UNIX)" on page 117)

Deprecated functionality

- INSTALL_TSAMP response file keyword (see "IBM Tivoli System Automation for Multiplatforms (SA MP) is now automatically installed" on page 112)
- The **db2_install** command (see "db2_install command has been deprecated (Linux and UNIX)" on page 146)

Discontinued functionality

- 32-bit client support (see "32-bit client support has been discontinued (HP-UX)" on page 156)
- Some operating systems (see "Some operating systems are no longer supported" on page 158)
- Control Center tools (see "Control Center tools have been discontinued" on page 160)
- The Worksheet Format (WSF) for input or output files in the **IMPORT** and **EXPORT** commands (see "Worksheet Format (WSF) for Import and Export utilities have been discontinued" on page 159)

- The **db2imigr** and **db2ckmig** commands (see “db2imigr and db2ckmig commands have been discontinued” on page 164)
- The **-s** parameter of the **db2iupdt** command (see “-s parameter of the db2iupdt command has been discontinued” on page 165)

Part 4. Appendixes

Appendix A. Functionality in DB2 features and DB2 product editions

Some functionality is available in only certain DB2 database product editions. In some cases, the functionality is associated with a particular DB2 feature.

The table indicates which functionality is included in a DB2 product edition. If the functionality is not included but it is available in a DB2 feature, the name of the feature is specified. You must obtain a license for that DB2 feature as well as for the DB2 database product edition.

Note: This table is for informational purposes only. For details of entitlement, rights and obligations, refer to the license agreement for your DB2 product.

Table 48. Functionality in DB2 features and DB2 database product editions

Functionality	DB2 Express-C	DB2 Express® Edition ¹	DB2 Workgroup Server Edition	DB2 Enterprise Server Edition ²	DB2 Advanced Enterprise Server Edition ²	IBM Database Enterprise Developer Edition
Advanced Copy Services	No	Yes	Yes	Yes	Yes	Yes
Adaptive Compression and classic row compression	No	No	No	DB2 Storage Optimization Feature ⁴	Yes	Yes
Compression: backup	Yes	Yes	Yes	Yes	Yes	Yes
Connection concentrator	No	No	No	Yes	Yes	Yes
Continuous Data Ingest	No	No	No	No	Yes	Yes
Database partitioning ³	No	No	No	No	No	No
functionality	No	No	No	No	No	Yes
DB2 Governor	No	No	No	Yes	Yes	Yes
DB2 pureScale functionality	No	No	You are entitled to use DB2 pureScale Feature in a maximum of 16 cores and 64 GB total cluster size.	DB2 pureScale Feature ⁴	DB2 pureScale Feature ⁴	Yes
Federation with DB2 LUW and Informix® Data Server data sources	Yes	Yes	Yes	Yes	Yes	Yes
Federation with DB2 LUW and Oracle data sources	No	No	No	No	Yes	Yes
High availability disaster recovery	No	Yes	Yes	Yes	Yes	Yes
IBM Data Studio	Yes	Yes	Yes	Yes	Yes	Yes
IBM InfoSphere Data Architect	No	No	No	No	Yes ⁵	Yes
IBM InfoSphere Optim Configuration Manager	No	No	No	No	Yes	Yes

Table 48. Functionality in DB2 features and DB2 database product editions (continued)

Functionality	DB2 Express-C	DB2 Express® Edition ₁	DB2 Workgroup Server Edition	DB2 Enterprise Server Edition ²	DB2 Advanced Enterprise Server Edition ²	IBM Database Enterprise Developer Edition
IBM InfoSphere Optim Performance Manager Extended Edition ³	No	No	No	No	Yes	Yes
IBM InfoSphere Optim pureQuery Runtime	No	No	No	No	Yes	Yes
IBM InfoSphere Optim Query Workload Tuner	No	No	No	No	Yes	Yes
Label-based access control (LBAC)	Yes	Yes	Yes	Yes	Yes	Yes
Materialized query tables (MQTs)	No	No	No	Yes	Yes	Yes
Multidimensional clustering (MDC) tables	No	No	No	Yes	Yes	Yes
Multi-Temperature Storage	No	No	No	Yes	Yes	Yes
Net Search Extender	Yes	Yes	Yes	Yes	Yes	Yes
Online reorganization	No	Yes	Yes	Yes	Yes	Yes
Oracle Compatibility	Yes	Yes	Yes	Yes	Yes	Yes
pureXML storage	Yes	Yes	Yes	Yes	Yes	Yes
Q Replication with two other DB2 LUW servers	No	No	No	No	Yes	Yes
Query parallelism	No	No	No	Yes	Yes	Yes
Replication tools	Yes ⁶	Yes ⁶	Yes ⁶	Yes ⁶	Yes ⁶	Yes
Row and column access control (RCAC)	Yes	Yes	Yes	Yes	Yes	Yes
Spatial Extender	Yes	Yes	Yes	Yes	Yes	Yes
SQL Replication between DB2 LUW and Informix Data Server	No	Yes	Yes	Yes	Yes	Yes
Sybase Compatibility	No	No	No	No	No	No
Table partitioning	No	No	No	Yes	Yes	Yes
Time Travel Query	Yes	Yes	Yes	Yes	Yes	Yes
Tivoli System Automation	No	Yes	Yes	Yes	Yes	Yes
Workload management	No	No	No	Yes	Yes	Yes

Note:

1. DB2 Express Edition including DB2 Express Edition Fixed Term License
2. You can purchase all of the DB2 features that are listed in this column for use with IBM InfoSphere Warehouse Enterprise Base and Enterprise Edition products.

3. DB2 Database Partitioning Feature (DPF) is no longer included in or available for any DB2 database editions. It is included in all IBM InfoSphere Warehouse product editions.
4. Separately priced feature.
5. DB2 Advanced Enterprise Server Edition includes 10 InfoSphere Data Architect user licenses.
6. Replication tools except the Replication Center are available on all supported operating systems. The Replication Center is available only on Linux and Windows operating systems.

Appendix B. Functionality in DB2 features in DB2 Connect product editions

Some functionality is available in only certain DB2 Connect product editions. In some cases, the functionality is associated with a particular DB2 feature.

The table indicates which functionality is included in a DB2 Connect product edition. If the functionality is not applicable to the DB2 Connect products, the value "Not applicable" is specified.

Table 49. Functionality in DB2 Connect product editions

Functionality	DB2 Connect Personal Edition	DB2 Connect server editions
Adaptive Compression	No	No
Advanced Copy Service	No	Yes
Compression: backup	No	No
Compression: Data	No	No
Compression: Index	No	No
Compression: Temp table	No	No
Compression: XML	No	No
Connection concentrator	No	Yes
Continuous Data Ingest	No	No
Database partitioning	No	No
DB2 Governor	No	Yes
Heterogeneous Federation	No	No
High availability disaster recovery	No	Yes
Homogeneous Federation	No	Yes
Homogeneous Q Replication	No	No
IBM Data Studio Version 3.1	Yes	Yes
IBM InfoSphere Optim Performance Manager Extended Edition ¹	No	No
IBM InfoSphere Optim pureQuery Runtime	No	Yes ²
Label-based access control (LBAC)	No	No
Materialized query tables (MQT)	No	Yes
Multidimensional clustering (MDC) tables	No	Yes
Multi-Temperature Storage	No	No
Online reorganization	No	No
DB2 pureScale	No	No
pureXML storage	No	No

Table 49. Functionality in DB2 Connect product editions (continued)

Functionality	DB2 Connect Personal Edition	DB2 Connect server editions
Query parallelism	No	Yes
Replication tools	No	Yes ³
Scan Sharing	No	No
Spatial Extender	No	Yes
Time Travel Query	Yes	Yes
Table partitioning	No	No
Tivoli System Automation	No	Yes
Workload management	No	Yes
<p>Note:</p> <ol style="list-style-type: none"> 1. IBM InfoSphere Optim Performance Manager Extended Edition is a follow-on to Performance Expert. IBM InfoSphere Optim Performance Manager Extended Edition helps optimize the performance and availability of mission-critical databases and applications. 2. Only DB2 Connect Unlimited Edition for System z and DB2 Connect Application Server Advanced Edition include IBM InfoSphere Optim pureQuery Runtime. 3. Replication tools except the Replication Center are available on all supported operating systems. The Replication Center is available only on Linux and Windows operating systems. 		

Related reference:

Appendix A, “Functionality in DB2 features and DB2 product editions,” on page 195

Appendix C. Overview of the DB2 technical information

DB2 technical information is available in multiple formats that can be accessed in multiple ways.

DB2 technical information is available through the following tools and methods:

- DB2 Information Center
 - Topics (Task, concept and reference topics)
 - 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](http://www.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 email 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 and translated DB2 Version 10.1 manuals in PDF format can be downloaded from www.ibm.com/support/docview.wss?rs=71&uid=swg2700947.

Although the tables identify books available in print, the books might not be available in your country or region.

The form number increases each time a manual is updated. Ensure that you are reading the most recent version of the manuals, as listed below.

Note: The *DB2 Information Center* is updated more frequently than either the PDF or the hard-copy books.

Table 50. DB2 technical information

Name	Form Number	Available in print	Last updated
<i>Administrative API Reference</i>	SC27-3864-00	Yes	April, 2012
<i>Administrative Routines and Views</i>	SC27-3865-00	No	April, 2012
<i>Call Level Interface Guide and Reference Volume 1</i>	SC27-3866-00	Yes	April, 2012
<i>Call Level Interface Guide and Reference Volume 2</i>	SC27-3867-00	Yes	April, 2012
<i>Command Reference</i>	SC27-3868-00	Yes	April, 2012
<i>Database Administration Concepts and Configuration Reference</i>	SC27-3871-00	Yes	April, 2012
<i>Data Movement Utilities Guide and Reference</i>	SC27-3869-00	Yes	April, 2012
<i>Database Monitoring Guide and Reference</i>	SC27-3887-00	Yes	April, 2012
<i>Data Recovery and High Availability Guide and Reference</i>	SC27-3870-00	Yes	April, 2012
<i>Database Security Guide</i>	SC27-3872-00	Yes	April, 2012
<i>DB2 Workload Management Guide and Reference</i>	SC27-3891-00	Yes	April, 2012
<i>Developing ADO.NET and OLE DB Applications</i>	SC27-3873-00	Yes	April, 2012
<i>Developing Embedded SQL Applications</i>	SC27-3874-00	Yes	April, 2012
<i>Developing Java Applications</i>	SC27-3875-00	Yes	April, 2012
<i>Developing Perl, PHP, Python, and Ruby on Rails Applications</i>	SC27-3876-00	No	April, 2012
<i>Developing User-defined Routines (SQL and External)</i>	SC27-3877-00	Yes	April, 2012
<i>Getting Started with Database Application Development</i>	GI13-2046-00	Yes	April, 2012

Table 50. 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>	GI13-2047-00	Yes	April, 2012
<i>Globalization Guide</i>	SC27-3878-00	Yes	April, 2012
<i>Installing DB2 Servers</i>	GC27-3884-00	Yes	April, 2012
<i>Installing IBM Data Server Clients</i>	GC27-3883-00	No	April, 2012
<i>Message Reference Volume 1</i>	SC27-3879-00	No	April, 2012
<i>Message Reference Volume 2</i>	SC27-3880-00	No	April, 2012
<i>Net Search Extender Administration and User's Guide</i>	SC27-3895-00	No	April, 2012
<i>Partitioning and Clustering Guide</i>	SC27-3882-00	Yes	April, 2012
<i>pureXML Guide</i>	SC27-3892-00	Yes	April, 2012
<i>Spatial Extender User's Guide and Reference</i>	SC27-3894-00	No	April, 2012
<i>SQL Procedural Languages: Application Enablement and Support</i>	SC27-3896-00	Yes	April, 2012
<i>SQL Reference Volume 1</i>	SC27-3885-00	Yes	April, 2012
<i>SQL Reference Volume 2</i>	SC27-3886-00	Yes	April, 2012
<i>Text Search Guide</i>	SC27-3888-00	Yes	April, 2012
<i>Troubleshooting and Tuning Database Performance</i>	SC27-3889-00	Yes	April, 2012
<i>Upgrading to DB2 Version 10.1</i>	SC27-3881-00	Yes	April, 2012
<i>What's New for DB2 Version 10.1</i>	SC27-3890-00	Yes	April, 2012
<i>XQuery Reference</i>	SC27-3893-00	No	April, 2012

Table 51. DB2 Connect-specific technical information

Name	Form Number	Available in print	Last updated
<i>DB2 Connect Installing and Configuring DB2 Connect Personal Edition</i>	SC27-3861-00	Yes	April, 2012
<i>DB2 Connect Installing and Configuring DB2 Connect Servers</i>	SC27-3862-00	Yes	April, 2012
<i>DB2 Connect User's Guide</i>	SC27-3863-00	Yes	April, 2012

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

Documentation for other versions of DB2 products is found in separate information centers on ibm.com[®].

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/>.

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 10.1 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 update existing Information Center features and languages. One benefit of automatic updates is that the Information Center is unavailable for a shorter time compared to during a manual update. In addition, automatic updates can be set to run as part of other batch jobs that run periodically.
- Manual updates can be used to update existing Information Center features and languages. Automatic updates reduce the downtime during the update process, however you must use the manual process when you want to add features or languages. 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. In the automatic update process the Information Center incurs an outage to restart the Information Center after the update only.

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/V10.1` 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 10.1` 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 by 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/db2icdv10 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 10.1* 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/V10.1` 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 that 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 `doc\bin` directory within the installation 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 `doc/bin` directory within the installation 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/db2icdv10 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 database 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/db2luw/v10r1/>.

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.

DB2 troubleshooting information

A wide variety of troubleshooting and problem determination information is available to assist you in using DB2 database products.

DB2 documentation

Troubleshooting information can be found in the *Troubleshooting and Tuning Database Performance* or the Database fundamentals section of the *DB2 Information Center*, which contains:

- Information about how to isolate and identify problems with DB2 diagnostic tools and utilities.
- Solutions to some of the most common problem.
- Advice to help solve other 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 are granted subject to the following terms and conditions.

Applicability: These terms and conditions are in addition to any terms of use for the IBM website.

Personal use: You may reproduce these publications for your personal, noncommercial 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.

Rights: 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.

IBM 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 www.ibm.com/legal/copytrade.shtml

Appendix D. 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 websites not owned by IBM 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, Celeron, 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 154

Numerics

32-bit client

HP-UX

discontinued support 156

A

activities monitor elements

new 34

Activity Monitor

discontinued 161

activity monitor routines

deprecated 140

administration

modifications summary 99

administrative routines

changed functionality 121

changes 121

agent priority

deprecated 141

alt_diagpath configuration parameter

enhancements 103

alt_diagpath database manager configuration parameter

changes 105

ALTER DATABASE

statement deprecated 151

ALTER EVENT MONITOR

overview 32

ALTER TABLE

COMPRESS clause changes 135

ALTER TABLESPACE

new clause 136

ALTER WORKLOAD statement

COLLECT UNIT OF WORK DATA clause changes 137

application development

changes summary 119

enhancements summary 73

archiving

log files

compression 49

automatic

deprecated 143

AUTOMATIC STORAGE parameter

deprecated 148

B

backups

enhancements summary 47

binary XML format 19

built-in functions

additions 125

changes 125

built-in global variables

introduction 73

built-in routines

additions 125

changes 125

built-in string functions

changed functionality 124

changes 124

built-in views

additions 125

changes 125

C

casting

XML casting errors 119

XML values

truncation 20

catalog views

additions 125

changes 125

cf_diagpath configuration parameter

enhancements 103

cf_diagpath database manager configuration parameter

changes 105

change history event monitor

overview 25

changed functionality

commands

db2cat 132

db2ckupgrade 132

db2cluster_prepare 133, 148

db2evtbl 133

db2icrt 133, 148

db2iupdt 133, 148

db2pd 135

commandsREORG INDEXES/TABLE 150

Replication Center 99

special registers 125

statements

ALTER TABLE 135

ALTER TABLESPACE 136

ALTER WORKLOAD 137

CREATE INDEX 138

CREATE TABLE 135

CREATE WORKLOAD 137

summary 97

changed names

thresholds 102

changes summary

application development 119

CLP (command line processor)

commands

changes 131

clustering

tables

insert time 14

COLLATE USING parameter

collation values deprecated 148

collations

language-aware

for Unicode data 95

- command
 - db2move 14
- Command Editor
 - discontinued 161
- command line processor (CLP)
 - command changes 131
- commands
 - db2cluster_prepare 81, 88
 - db2icrt
 - enhancements 81, 88
 - db2iupdt
 - enhancements 81, 88
 - db2prereqcheck 83
 - db2setup 81, 88
 - db2val 81, 88
 - discontinued
 - Control Center 161
 - dynexpln
 - deprecated 147
 - installFixPack
 - enhancements 81, 88
 - modifications summary 131
 - PRUNE LOGFILE
 - deprecated 147
- commands deprecated
 - db2_install 146
- comparisons
 - XML 20
- composite indexes
 - jump scan 59
- compression 9
- Configuration Assistant (CA)
 - discontinued 161
- configuration changes
 - tracking 25
- configuration parameters
 - additions 105, 112
 - changes 105, 112
 - new 105, 112
- Control Center
 - discontinued tools 161
 - extensions
 - discontinued 161
- Control Center tools
 - discontinued 161
- CREATE DATABASE
 - parameter values deprecated 148
 - parameters deprecated 148
- CREATE INDEX statement
 - DETAILED keyword changes 138
- CREATE TABLE
 - COMPRESS clause changes 135
- CREATE WORKLOAD statement
 - COLLECT UNIT OF WORK DATA clause changes 137
- created temporary tables
 - implicitly hidden columns 121
- CURRENT MEMBER
 - DB2 pureScale improvements 92
 - details 87

D

- data partitions
 - adding
 - enhanced data accessibility 15
 - attaching
 - enhanced data accessibility 15

- data recovery
 - log replay delay 49
- Data Studio
 - overview 85
- data tag class thresholds
 - new monitor elements 34
- data types
 - declared 64
- database configuration changes
 - tracking 25
- database configuration parameters
 - changes 112
 - deprecated 152, 168
 - discontinued 152, 168
 - new 112
 - overview 112
- database manager configuration changes
 - tracking 25
- database manager configuration parameters
 - changes 105
 - deprecated 152, 168
 - discontinued 152, 168
 - new 105
- databases
 - setup changes summary 111
- DATATAGINSC activity threshold
 - overview 68
- DB2 APIs
 - database history records 164
- DB2 Connect 199
 - changes summary 189
 - DB2 changes summary 191
 - DB2 enhancements summary 191
 - enhancements summary 189
- DB2 features
 - overview 195
- DB2 Geodetic Data Management Feature
 - discontinued 163
- DB2 Information Center
 - updating 204, 206
 - versions 204
- DB2 JDBC Type 2 Driver
 - discontinued 156
- DB2 products
 - availability 7
 - functionality by edition 195
 - packaging 7
- DB2 pureScale
 - AIX
 - RDMA over Converged Ethernet 88
 - installing
 - included in other installations 84, 87
- DB2 Spatial Extender
 - integrated installation 84
- DB2 system commands
 - modifications summary 131
- DB2 Text Search
 - administrative commands and stored procedures
 - changes 101
 - scheduler changes 101
 - searching improvements 79
 - security model changes 100
 - stand-alone server deployment 79
 - support for partitioned database environments 80
 - support for partitioned tables 80
 - text index location change 101

- DB2 workload management
 - enhancements
 - summary 65
 - WLM dispatcher 65
 - overview 69, 93
 - predictive prioritization
 - overview 68
 - reactive prioritization
 - overview 68
- db2_install command
 - deprecated 146
- DB2_LIKE_VARCHAR registry variable
 - deprecated functionality 151
- DB2_NO_FORK_CHECK registry variable
 - changes 107
- db2cat command
 - command output changes 132
- db2ckmig command
 - discontinued 164
- db2ckupgrade command
 - default behaviour changed 132
- db2cluster_prepare command
 - changes 133, 148
 - enhancement 81, 88
- db2evtbl command
 - changes 133
- db2icrt command
 - changes 133, 148
 - enhancements 81, 88
- db2IdentifyType1 command
 - deprecated command 145
- db2imigr command
 - discontinued 164
- db2iupdt command
 - s parameter discontinued 165
 - changes 133, 148
 - enhancements 81, 88
- db2move command 14
- DB2NTNOCACHE registry variable
 - changes 107
- db2pd
 - tablespaces parameter change 135
- db2prereqcheck command 83
- DB2SE_USA_GEOCODER
 - discontinued 157
- db2setup command
 - enhancement 81, 88
- db2support command
 - enhancements 93
- db2val command
 - DB2 pureScale 91
 - details 87
 - enhancement 81, 88
- DECIMAL data type
 - indexes over XML 17
- declared temporary tables
 - implicitly hidden columns 121
- deprecated functionality
 - administrative views
 - SNAPHADR 153
 - agent priority 141
 - automatic 143
 - collations
 - UCA400_LSK 142
 - UCA400_LTH 142
 - UCA400_NO 142
- deprecated functionality (*continued*)
 - command parameter values
 - COLLATE USING 148
 - command parameters
 - flushbp 149
 - AUTOMATIC STORAGE 148
 - COLLATE USING 142
 - CREATE DATABASE 148
 - MANAGED BY SYSTEM 148
 - commands
 - file parameter of the db2flsn and db2rfpen
 - commands 165
 - db2_install 146
 - db2IdentifyType1 145
 - dynexpln 147
 - PRUNE LOGFILE 147
 - database configuration parameters 152, 168
 - database manager configuration parameters 152, 168
 - monitor elements
 - hadr_connect_status 153
 - hadr_connect_time 153
 - hadr_heartbeat 153
 - hadr_local_host 153
 - hadr_local_service 153
 - hadr_log_gap 153
 - hadr_peer_window 153
 - hadr_peer_window_end 153
 - hadr_primary_log_file 153
 - hadr_primary_log_lsn 153
 - hadr_primary_log_page 153
 - hadr_remote_host 153
 - hadr_remote_instance 153
 - hadr_remote_service 153
 - hadr_role 153
 - hadr_standby_log_file 153
 - hadr_standby_log_lsn 153
 - hadr_standby_log_page 153
 - hadr_state 153
 - hadr_syncmode 153
 - hadr_timeout 153
 - Net Search Extender 145
 - SMS
 - USER TABLESPACES 142
 - statements
 - ALTER DATABASE 151
 - summary 97, 139, 171
 - table functions
 - activity monitor routines 140
 - SNAP_GET_HADR 153
 - version-suffixed SQL administrative routines 144
- deprecated parameters
 - db2pdcfg command
 - flushbp parameter 149
- dft_schemas_dcc database configuration parameter 50
 - overview 112
- diagpath configuration parameter
 - enhancements 103
- diagpath database manager configuration parameter
 - changes 105
- discontinued functionality
 - 32-bit client support
 - HP-UX 156
 - commands
 - db2ckmig 164
 - db2imigr 164
 - db2iupdt -s parameter 165
- Control Center tools 161

- discontinued functionality (*continued*)
 - database configuration parameters 152, 168
 - logretain 152, 168
 - user exit 152, 168
 - database manager configuration parameters 152, 168
 - DB2 APIs
 - database history records 164
 - DB2 Geodetic Data Management Feature 163
 - DB2 JDBC Type 2 Driver 156
 - DB2SE_USA_GEOCODER 157
 - Microsoft Systems Management Server 157
 - Query Patroller 160
 - summary 97, 155, 171
 - Systems Management Server 157
 - version-suffixed SQL administrative routines 158
 - Visual Studio 2005 159
- discontinued parameters
 - db2flsn command
 - deprecated 165
 - db2rfpen command
 - deprecated 165
- documentation
 - overview 201
 - PDF files 201
 - printed 201
 - terms and conditions of use 208
- dynexpln
 - command deprecated 147

E

- environment variables
 - changes 107
- Event Analyzer
 - discontinued 161
- event monitor tables
 - upgrade from previous releases 33
- event monitors
 - change history
 - overview 25
 - locking
 - output options 31
 - output options
 - overview 31
 - package cache
 - output options 31
 - UE tables
 - pruning data 34
 - unit of work
 - output options 31
- executable IDs
 - unit of work event monitor enhancement 31
- explain facility
 - output
 - changes 103, 134
 - procedure changes 134
- explain tables
 - EXPLAIN_OBJECT change 134
- EXPLAIN_OUTPUT table
 - change 134

F

- fix packs
 - enhancements summary 81

- fn:exists function
 - indexes 17
- fn:starts-with function 120
 - indexes 17
- fn:upper-case function
 - indexes 17
- functions
 - additions 125
 - changes 125
 - generic table functions
 - overview 74

G

- Geodetic Data Management Feature
 - discontinued 163
- Geodetic Extender
 - discontinued 163
- global variables
 - additions 125
 - built-in 73
 - changes 125

H

- HADR multiple standbys
 - overview 47
- HADR standby
 - log spooling 49
- hadr_replay_delay database configuration parameter 49
 - overview 112
- hadr_spool_limit database configuration parameter 49
- Health Center
 - discontinued 161
- help
 - SQL statements 204
- high availability
 - enhancements
 - controlling automatic failback 91
 - db2cluster command 91
 - repairing instance domain 91
 - enhancements summary 47
- highlighting conventions xi
- HP-UX
 - 32-bit client support discontinued 156

I

- I/O servers
 - new monitor elements 34
- IBM data server clients
 - enhancements 76
- IBM data server drivers
 - enhancements 76
- implicitly hidden columns
 - created temporary tables 121
 - declared temporary tables 121
- Import
 - Export
 - utilities 159
- improvements 9
 - storage management 11
- in-service-class thresholds
 - DATATAGINSC 68
- index gaps
 - jump scan 59

- indexes 120
 - XML
 - case-insensitive searches 17
 - DECIMAL 17
 - functional 17, 120
 - INTEGER 17
- Indoubt Transaction Monitor
 - discontinued 161
- insert time clustering tables
 - new functionality 14
- insert time-clustered tables
 - reclaim unused space 13
- installation
 - changes summary 111
 - DB2 product media
 - DB2 Spatial Extender 84
 - DB2 pureScale
 - included in other installations 84, 87
 - enhancements summary 81
- installFixPack command
 - details 87
 - enhancement 81, 88
- INTEGER data type
 - indexes over XML 17
- intra-partition parallelism
 - enhancements 55

J

- JDBC
 - enhancements summary 76
- joins
 - star schema 60
 - zigzag 60
- Journal
 - discontinued 161

K

- Kerberos authentication protocol
 - What's changed 117

L

- License Center
 - discontinued 161
- license policies
 - enhancements 83
- locales
 - added in Version 10.1 96
- log spooling
 - HADR configuration 49
- logical data groups
 - event monitors
 - changing 32
- logretain database configuration parameter
 - discontinued 152, 168
- logs
 - archiving
 - compression 49
 - enhancements summary 47

M

- manageability
 - enhancements summary 9
- MANAGED BY SYSTEM
 - CREATE DATABASE
 - deprecated 148
 - USER TABLE SPACES
 - deprecated 142
- managing instances
 - changes 133, 148
- memory
 - allocating
 - POWER7 57
- Memory Visualizer
 - discontinued 161
- Microsoft Systems Management Server
 - discontinued 157
- modules
 - features 195
- MON_GET_GROUP_BUFFERPOOL
 - DB2 pureScale 93
- mon_obj_metrics database configuration parameter
 - changes 105
- mon_uow_data database configuration parameter
 - changes 112
- mon_uow_execlist database configuration parameter
 - overview 112
- mon_uow_pkglist database configuration parameter
 - overview 112
- monitoring
 - enhancements 25
 - functions 27
 - views 27
 - summary 25
- multi-temperature data storage
 - overview 10
- multicultural support
 - enhancements summary 95
- multiple storage groups 11

N

- Net Search Extender
 - deprecated 145
- networks
 - cluster interconnect
 - RDMA over Converged Ethernet 88
- new features
 - highlights 3
 - summary 1
- non-nested activities
 - new monitor elements 34
- notices 211
- numdb database manager configuration parameter
 - changes 105

O

- operators
 - REBAL 55
- optimization profiles
 - enhancements 54

P

- package listing
 - unit of work event monitor enhancements 99
- partitioned tables
 - adding data partitions
 - enhanced data accessibility 15
 - attaching data partitions
 - enhanced data accessibility 15
- performance
 - enhancements
 - summary 51
- performance enhancements
 - ingest utility 48
- prefetching
 - smart data prefetching 57
 - smart index prefetching 57
- problem determination
 - information available 208
 - tutorials 208
- procedures
 - declared 64
- product availability and packaging 199
- prune data
 - UE tables
 - event monitor 34
- PRUNE LOGFILE
 - command deprecated 147

Q

- queries
 - performance
 - enhancement 51, 55
- Query Patroller
 - discontinued 160

R

- range partitioning
 - DB2 pureScale 90
 - RDF
 - what's new 75
 - REBAL operator 55
 - reclaim index space
 - improvements 13
 - reclaim table space
 - improvements 13
 - recovery
 - enhancements summary 47
 - registry variables
 - changes 107
 - DB2_RESOURCE_POLICY
 - enhancements 57
 - deprecated functionality 151
 - discontinued support 166
 - new 107
 - REORG INDEXES/TABLE
 - changed 150
 - Replication Center
 - changes 99
 - resiliency
 - enhancements summary 47
 - response files
 - keywords
 - addition 83
 - routines
 - additions 125
 - changes 125
 - RUNSTATS command
 - index sampling 53
 - RUNSTATS utility
 - index sampling 53
- ## S
- Satellite Administration Center
 - discontinued 161
 - scalar functions
 - additions for monitoring 27
 - INSTRB 64
 - TIMESTAMPDIFF 64
 - TO_SINGLE_BYTE 64
 - schema replication 50
 - security
 - changes summary 117
 - data 71
 - enhancements summary 71
 - FGAC 71
 - Fine-grained access control 71
 - RCAC 71
 - Row and column access control 71
 - service classes
 - agent priority
 - deprecated 141
 - silent installation
 - keywords 83
 - SMS
 - USER TABLE SPACES
 - deprecated 142
 - SNAP_GET_HADR table function
 - deprecated 153
 - SNAPHADR administrative view
 - deprecated 153
 - Spatial Extender
 - integrated installation 84
 - special registers
 - changes 125
 - SQL
 - compatibility enhancements 63
 - SQL administrative routines
 - deprecated 144
 - discontinued 158
 - SQL statements
 - changes 131
 - help
 - displaying 204
 - performance improvements 51
 - SQL16061N 119
 - statement keys
 - enhancements 54
 - statement metrics
 - unit of work event monitor enhancement 31
 - statement thresholds 26
 - statements
 - ALTER NICKNAME
 - NOT ENFORCED changes 111
 - ALTER TABLE
 - NOT ENFORCED changes 111
 - CREATE NICKNAME
 - NOT ENFORCED changes 111
 - CREATE TABLE
 - NOT ENFORCED changes 111

- statistical views
 - query optimizer enhancements
 - overview 55
- storage groups
 - multiple 11
 - overview 10
- storage management
 - improvements 11
- SYSCAT views
 - additions 125
 - changes 125
- system catalog views
 - changed functionality 121
 - changes 121
- system catalogs
 - views
 - additions 125
 - changes 125
- Systems Management Server (SMS)
 - discontinued 157
- sysptime_period_adj database configuration parameter
 - overview 112

T

- table functions
 - monitoring
 - additions 27
 - changes 27
- table spaces
 - storage group enhancement 12
- tables
 - clustering
 - insert time 14
 - temporal 75
- target tables
 - event monitor
 - upgrade from previous releases 33
- Task Center
 - discontinued 161
- temporal tables
 - overview 75
 - Time Travel Query 75
- terms and conditions
 - publications 208
- territory codes
 - added in Version 10.1 96
- Text Search Extender
 - enhancements summary 79
- thresholds
 - changes 102
 - DATATAGINSC 68
 - domain
 - statement 26
- Time Travel Query
 - temporal tables 75
- time-spent monitor elements
 - new 34
- Tivoli System Automation for Multiplatforms (SA MP)
 - automatic installation 112, 142
- tools 154
- triggers
 - multiple-event 63
- troubleshooting 154
 - DB2 pureScale
 - improvements 93
 - online information 208

- troubleshooting (*continued*)
 - split diagpath 103
 - tutorials 208
- tutorials
 - list 207
 - problem determination 208
 - pureXML 207
 - troubleshooting 208

U

- UCA400_ISK
 - collation deprecated 142
- UCA400_LTH
 - collation deprecated 142
- UCA400_NO
 - collation deprecated 142
- UE tables
 - event monitors
 - pruning data 34
- unformatted event tables
 - upgrade from previous releases 33
- Unicode
 - string comparison 95
- Unicode databases
 - changed functionality 124
 - changes 124
- unit of work event monitor
 - data collection
 - executable ID list enhancement 31
 - package list enhancement 99
- updates
 - DB2 Information Center 204, 206
- upgrade
 - event monitor tables 33
- upgrades
 - enhancements summary 81
- usage lists
 - overview 26
- userexit
 - discontinued 152, 168

V

- version-suffixed SQL administrative routines
 - deprecated 144
 - discontinued 158
- views
 - additions 125
 - changes 125
 - monitoring enhancements 27
- Visual Studio 2005
 - discontinued 159

W

- What's new
 - summary 1
- wlm_disp_concur database manager configuration parameter
 - overview 105
- wlm_disp_cpu_shares database manager configuration parameter
 - overview 105
- wlm_disp_min_util database manager configuration parameter
 - overview 105

- wlm_dispatcher database manager configuration parameter
 - overview 105
- workloads
 - new monitor elements 34
- Worksheet Format
 - WSF 159
- write to tables
 - upgrade from previous releases 33

X

- XDBX format 19
- XML
 - case-insensitive searches 17
 - casting errors 119
 - enhancements
 - summary 17
 - functional indexes 17, 120
- XML casting errors 119
- XMLCAST
 - truncation 20
- xmlFormat property 19
- XMLTABLE function
 - casting 20
 - performance 22



Printed in USA

SC27-3890-00



Spine information:

IBM DB2 10.1 for Linux, UNIX, and Windows

What's New for DB2 Version 10.1

