Db2 11 for z/OS

What's New?



GC19-4068-11

Notes

Before using this information and the product it supports, be sure to read the general information under "Notices" at the end of this information.

Subsequent editions of this PDF will not be delivered in IBM Publications Center. Always download the latest edition from PDF format manuals for Db2 11 for z/OS (Db2 Home).

2020-07-31 edition

This edition applies to Db2® 11 for z/OS® (product number 5615-DB2), Db2 11 for z/OS Value Unit Edition (product number 5697-P43), and to any subsequent releases until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

Specific changes are indicated by a vertical bar to the left of a change. A vertical bar to the left of a figure caption indicates that the figure has changed. Editorial changes that have no technical significance are not noted.

© Copyright International Business Machines Corporation 2013, 2019.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this information..... vii
Terminology and citations..... vii
How to send your comments..... viii

Part 1. What's new in Db2 11..... 1

Chapter 1. Recent enhancements in Db2 11..... 3
Improved reliability and contention relief for online migration (APAR PH10826 - May 2020)..... 3
New job DSNTIJPM premigration report 23: SQL routines created in DB2 9 (PH07660 - November 2019)..... 3
Ability to control workload balancing of accelerated queries based on accelerator version (PH05477 - September 2019)..... 3
New job DSNTIJPM premigration report 24: active log data sets larger than 4 GB (PH11514 - July 2019)..... 4
Multiple captures on a single proxy for GDPS Continuous Availability with zero data loss (PI80787 - June 2019)..... 4
New environment variable to control the MQListener log size (PH07230 - March 2019)..... 4
Avoid external sorts for single-column column groups at the utility job level (PH03678 - February 2019)..... 4
User-defined function with MODIFIES SQL DATA from a subselect (PI93887 - January 2019)..... 5
IBM zHyperLink: database synchronous read I/O exploitation (PH01123 - December 2018)..... 5
Allow ROTATE PARTITION for materialized query tables or tables having dependent materialized query tables (PH00194 - October 2018)..... 5
Db2 command line processor (CLP) support for user name and credentials for running SQL files (PH00335 - September 2018)..... 5
Suppress SQL warning messages for OPEN and FETCH from SPUFI, DSNTEP2, DSNTEP4, and DSNTIAUL (PI79053 - June 2018)..... 6
Profiles can start automatically when Db2 starts (PI89912 - January 2018)..... 6
Automatic remigration binds are removed in release coexistence and fallback (PI87675 - January 2018)..... 6
DSNTIJSG installation job no longer creates SYSIBM EXPLAIN tables (PI86450 - December 2017)..... 7
MQListener automatic restart after Db2 restart (PI84698 - December 2017)..... 7
LOAD RESUME SUPPORT FOR INLINE COPY (PI81723 - December 2017)..... 7
Hybrid Transactional Analytical Processing support for accelerated queries (PI83286 and PI83288 - November 2017)..... 7
z/OS DFSMS data set encryption (PI81900 - November 2017)..... 8
Support for collection of more accurate real storage usage statistics (PI82191 - August 2017)..... 8
Explicitly-defined hidden ROWID columns (PI77719 - June 2017)..... 8
New LOAD utility IGNORE options (PI79716 - June 2017)..... 9
ADMIN_INFO_SQL stored procedure enhanced to collect UDF information (PI45103 - April 2017)..... 9
LOAD utility support for more date and time formats (PI68139 - November 2016)..... 9
Reduced cost for collecting Db2 frequency statistics (PI74408 - August 2017)..... 11
LOAD REPLACE SHRLEVEL REFERENCE (PI67793 - November 2016)..... 11

Chapter 2. Performance..... 13
Support for indexes that exclude NULL keys..... 13
Improved performance of stored procedure calls from ODBC applications..... 13
Improved performance of ODBC applications that retrieve LOB or XML objects..... 13

Buffer pool enhancements....14
Performance improvements for the LOAD utility....14
Performance improvements for the REORG TABLESPACE utility....15
Improvements for RELEASE(DEALLOCATE) processing....15
Faster recovery of catalog and directory objects....16
Reduction in CPU usage for large numbers of partitions....16
CPU reduction for distributed applications....16
Improved performance for ROLLBACK TO SAVEPOINT....16
Db2 DPSI performance enhancements....17
Reductions in synchronous log writes....17
Relief from storage shortages in group buffer pools....17
Automatic cleanup of pseudo-empty index pages....18
Improved duplicate removal from SQL results....18
Performance improvements for distributed applications....19
Expanded RBA and LRSN log records....19
Improved access path reuse with warnings....19
Free space management enhancements....20
Improved predicate transformations....20
Predicate selectivity overrides....21

Chapter 3. Availability....23
Automatic recovery of indexes from GRECP or LPL status....23
Online schema enhancements....23
Improved data availability when altering limit keys....24
Automated REORG mapping table management....25
Work file database enhancements....25
Governing of parallel processing of utilities....25
Improved availability of previous compression dictionaries....26
Db2 changes for the GDPS Continuous Availability with zero data loss solution....26

Chapter 4. Ease of use....29
Improved data validation after running DSN1COPY....29
Ability to alter limit keys for materialized query tables....30
Improvements to DROP PENDING CHANGES....30
Support for archive tables....30
Optimization statistics feedback....31
Statistics collection enhancements....31
Setting special registers with a profile....32

Chapter 5. Security and regulatory compliance....33
Db2 enhancements for exit authorization checking....33
Db2 enhancements for program authorization....33
Auditing enhancements....34

Chapter 6. New application features....35
Autonomous procedures....35
Support for explicit period specification on views....35
Improvements to querying temporal tables....36
XML enhancements....37
XML support for the cross-loader function....37
Support for Java stored procedures in a 64-bit JVM....37
Support for variable-length timestamps in ODBC applications....37
Support for arrays....38
Expanded support for not-logged table spaces....38
Enhancement to the LIKE predicate....38
Support for user-defined global variables....38
Column masking improvements....39
Transform XML with an XSL style sheet....39

RRSAF connection functions that support longer client information fields..... 39
Support for temporal auditing.....40
DevOps with Db2: Automated deployment of applications with IBM UrbanCode Deploy..... 40
Modern language support Db2 for z/OS application development.....40
Db2 REST services improve efficiency and security..... 40

Chapter 7. Migration..... 43
Improved reliability and contention relief for online migration.....43
Improvements to Db2 installation and samples.....43
Application compatibility by version.....44

Part 2. What's changed in Db2 11.....45

Chapter 8. Incompatible changes for migration to Db2 11.....47
Application and SQL release incompatibilities.....47
V10R1 application compatibility level.....61
Utility release incompatibilities..... 65
Command release incompatibilities..... 70
Storage release incompatibilities..... 72
Other release incompatibilities.....72

Chapter 9. Command changes in Db2 11..... 79

Chapter 10. Utility changes in Db2 11..... 83

Chapter 11. SQL changes in Db2 11.....87

Chapter 12. Catalog changes in Db2 11.....93

Chapter 13. EXPLAIN table changes in Db2 11..... 99

Chapter 14. IFCID changes in Db2 11..... 101

Chapter 15. New and changed subsystem parameters in Db2 11..... 105

Chapter 16. The extended 10-byte RBA and LRSN in Db2 11..... 111

Chapter 17. Function that Db2 11 no longer supports.....115

Chapter 18. Deprecated function in Db2 11.....119

Information resources for Db2 11 for z/OS and related products..... 125

Notices.....127
Programming interface information.....128
Trademarks.....128
Terms and conditions for product documentation.....128
Privacy policy considerations.....129

Glossary.....131

Index..... 133

About this information

This information provides an executive overview of new function in Db2 11 for z/OS. The topics in this information provide a framework for describing new function in Db2 for z/OS. New functions are categorized according to user benefits such as information on demand, availability, and performance.

In addition, this information summarizes changes that were introduced in this version for Db2 commands, Db2 utilities, SQL statements, the Db2 catalog, Db2 performance monitoring, and instrumentation facility component identifiers (IFCIDs).

Throughout this information, "Db2" means "Db2 11 for z/OS". References to other Db2 products use complete names or specific abbreviations.

Important: To find the most up to date content for Db2 11 for z/OS, always use IBM® Knowledge Center or download the latest PDF file from PDF format manuals for Db2 11 for z/OS (Db2 Home).

This information assumes that Db2 11 is running in new-function mode, and that your application is running with the application compatibility value of 'V11R1'.

Availability of new function in Db2 11

The behavior of data definition statements such as CREATE, ALTER, and DROP, which embed data manipulation SQL statements that contain new capabilities, depends on the application compatibility value that is in effect for the application. An application compatibility value of 'V11R1' must be in effect for applications to use new capability in embedded statements such as SELECT, INSERT, UPDATE, DELETE, MERGE, CALL, and SET *assignment-statement*. Otherwise, an application compatibility value of 'V10R1' can be used for data definition statements.

Generally, new SQL capabilities, including changes to existing language elements, functions, data manipulation statements, and limits, are available only in new-function mode with applications set to an application compatibility value of 'V11R1'.

Optimization and virtual storage enhancements are available in conversion mode unless stated otherwise.

SQL statements can continue to run with the same expected behavior as in DB2® 10 new-function mode with an application compatibility value of 'V10R1'.

Who should read this information

This information is written primarily for people who are evaluating and planning for Db2 for z/OS.

Terminology and citations

When referring to a Db2 product other than Db2 for z/OS, this information uses the product's full name to avoid ambiguity.

About the Db2 brand change: IBM has re-branded DB2 to Db2, and Db2 for z/OS is now the name of the offering previously know as "DB2 for z/OS". During and after implementation of this change, you might sometimes still see references to the original names, such as "DB2 for z/OS" and "DB2", in different IBM web pages and documents. However, assume that these refer to the same "Db2 for z/OS" product when the PID, Entitlement Entity, version, modification, and release information match. For more information, see Revised naming for IBM Db2 family products on IBM z/OS platform.

The following terms are used as indicated:

Db2
Represents either the Db2 licensed program or a particular Db2 subsystem.

Tivoli® OMEGAMON® XE for Db2 Performance Expert on z/OS
Refers to any of the following products:

© Copyright IBM Corp. 2013, 2019

- IBM Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS
- IBM Db2 Performance Monitor on z/OS
- IBM Db2 Performance Expert for Multiplatforms and Workgroups
- IBM Db2 Buffer Pool Analyzer for z/OS

C, C++, and C language
Represent the C or C++ programming language.

CICS®
Represents CICS Transaction Server for z/OS.

IMS
Represents the IMS Database Manager or IMS Transaction Manager.

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

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

How to send your comments

Your feedback helps IBM to provide quality information. Please send any comments that you have about this book or other Db2 for z/OS documentation.

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

Part 1. What's new in Db2 11

Db2 11 for z/OS (also referred to as Db2 11 or Db2) delivers key innovations that reduce your total cost of ownership. Db2 11 provides enhanced analytics and increased availability, reliability, and security for your business-critical information. In addition, Db2 11 offers improvements that make installation and migration simpler and faster.

Performance improvements

Performance improvements in Db2 11 focus on optimizing query processing and reducing CPU processing time without causing significant administration or application changes. However, Db2 11 also offers a balanced approach to performance improvements across all types of workloads, whether your workloads are for online transaction processing (OLTP), batch, or utilities.

For more information about performance improvements, see Performance.

Expanded RBA and LRSN log records

In Db2 11, the relative byte address (RBA) and log record sequence number (LRSN) log records are expanded from basic 6-byte format to extended 10-byte format. You also can convert the RBA and LRSN to extended 10-byte format to avoid reaching the logging limits. Extending the RBA and LRSN to 10-byte format helps to avoid wrapping of the RBA and LRSN values, which can cause system problems. In addition, the increased precision of the 10-byte format for the LRSN provides performance improvements for data sharing environments.

For more information about this enhancement, see Expanded RBA and LRSN log records.

Availability enhancements

In Db2 11 you can alter the limit keys for a partitioned table space without impacting the availability of the data. When you change the limit key values, the data remains available and applications can continue to access the data.

For more information about this and other availability improvements, see Improved availability when altering limit keys and Availability.

Archive transparency

In Db2 11, Db2 can automatically insert rows that are deleted from one table into a separate table that is called an *archive table*. Archive tables provide the following benefits:

- Db2 can manage historical data for you. You do not have to manually move data to a separate table.
- Because rows that are infrequently accessed are stored in a separate table, you can potentially improve the performance of queries against the archive-enabled table.
- You can modify queries to include or exclude archive table data without having to change the SQL statement and prepare the application again. Instead, you can control the scope of the query with a global variable.
- You can store archive tables on a lower-cost device to reduce operating costs.

For more information about archive tables, see Support for archive tables.

For information about other ease of use enhancements in Db2 11, see Ease of use.

Security and regulatory compliance

With key enhancements in Db2 11, Db2 for z/OS and IBM Z continue to lead the industry in security and auditing.

For more information about security enhancements, see Security and regulatory compliance.

© Copyright IBM Corp. 2013, 2019

New application features

In Db2 11, expanded support for SQL, XML, and temporal tables results in improved application performance.

For more information about expanded application features, see New application features.

Simpler, faster migration

In Db2 11, simpler, faster migration results in a faster return on your investment. This version of Db2 for z/OS provides enhancements to the Db2 installation CLIST, ISPF panels, and jobs, and provides new installation verification procedures (IVPs). Also, a new feature helps to streamline the migration process by allowing an application with incompatible SQL or XML to continue running on Db2 11 without requiring code changes. You no longer need to wait for application changes to be planned and delivered for your business to realize the benefits of Db2 11.

For more information about migration enhancements, see Migration.

Related concepts

What's changed in Db2 11
Use this information when you are planning migration to Db2 11 and for planning to adopt new capabilities that Db2 11 introduces.

Related information

Db2 11 for z/OS Technical Overview (IBM Redbooks)

Chapter 1. Recent enhancements in Db2 11

A number of other enhancements to Db2 11 after general availability are introduced through APARs.

The enhancements described here are listed roughly in order of availability, beginning with the most recent.

Improved reliability and contention relief for online migration (APAR PH10826 - May 2020)

This APAR introduces the capability to identify applications that might be incompatible with catalog migrations and updates.

Before scheduling an online migration, you can issue the new DISPLAY BLOCKERS command or invoke the new BLOCKING_THREADS built-in function to identify locks and claims that might interfere with migration processing. This information allows you to take appropriate actions beforehand to minimize the possibility of a failed migration.

The new command and function can be used for both system and non-system databases. For more information, see Identify applications that are incompatible with online migration (Db2 Installation and Migration).

Related reference

-DISPLAY BLOCKERS (Db2) (Db2 Commands)

BLOCKING_THREADS (Db2 SQL)

New job DSNTIJPM premigration report 23: SQL routines created in DB2 9 (PH07660 - November 2019)

APAR PH07660 adds report 23 to the premigration job DSNTIJPM. Report 23 identifies native SQL procedures and compiled SQL scalar functions created in DB2 9. Such routines are subject to automatic binds after migration to Db2 12. You can use the report to regenerate these routines before migration to avoid performance problems that can result from too many concurrent automatic bind service tasks.

Related tasks

Rebind old plans and packages Db2 10 to avoid disruptive autobinds in Db2 11 (Db2 Installation and Migration)

Run premigration queries (DSNTIJPM) (Db2 Installation and Migration)

Ability to control workload balancing of accelerated queries based on accelerator version (PH05477 - September 2019)

In a Db2 for z/OS environment that includes a mixture of IBM Db2 Analytics Accelerator for z/OS V5 and V7 accelerator servers, PH05477 enables the ability to control workload balancing such that queries can be accelerated to either a V5 accelerator or a V7 accelerator based on workload volume. Prior to PH05477, workload balancing was limited to a single version of an accelerator.

To control this workload balancing behavior, a new option (10 - BALANCE WORKLOAD BETWEEN VERSIONS) has been added to the QUERY_ACCEL_OPTIONS subsystems parameter.

Related reference

ACCELERATION OPTIONS field (QUERY_ACCEL_OPTIONS subsystem parameter) (Db2 Installation and Migration)

© Copyright IBM Corp. 2013, 2019

New job DSNTIJPM premigration report 24: active log data sets larger than 4 GB (PH11514 - July 2019)

APAR PH11514 adds report 24 the to the premigration job DSNTIJPM. Report 24 identifies active log data sets greater than 4 GB in size. Before function level 500 is activated in Db2 12, the maximum supported active log data set size is 4 GB. Consequently, during migration to Db2 12, if the size of an active log data set in Db2 11 is greater than 4 GB, Db2 12 abnormally terminates during startup and issues message DSNJ158I. You can use the report to avoid this situation.

Related concepts

Active log data sets storage requirements (Db2 Installation and Migration)

Related tasks

Run premigration queries (DSNTIJPM) (Db2 Installation and Migration)

Related information

DSNJ158I (Db2 Messages)

Multiple captures on a single proxy for GDPS Continuous Availability with zero data loss (PI80787 - June 2019)

The GDPS® Continuous Availability with zero data loss solution provides disaster recovery with continuous application availability for enterprise businesses using the z/OS operating system. When Db2 is in a GDPS Continuous Availability with zero data loss environment, IFI applications issue READS calls for IFCID 0306 to a proxy data sharing group, and the proxy data sharing group captures log records from a source data sharing group.

Before this APAR, the GDPS Continuous Availability with zero data loss solution supported only a single IFCID 306 reader on the proxy member. To minimize the capture latency during a maintenance window, Db2 is enhanced to support multiple IFCID 306 readers on the proxy.

Related tasks

Reading complete log data for the GDPS Continuous Availability with zero data loss solution (Db2 Administration Guide)

New environment variable to control the MQListener log size (PH07230 - March 2019)

This APAR lets you control the maximum size of the MQListener HFS log file using new environment variable MQLSNLWR.

Before this APAR, the HFS log file into which MQListener writes diagnostic information could grow until it reached the HFS limit. For many users, that file size was much too big. After this APAR is applied, users can specify the maximum size for the log file. MQListener creates a copy of and reinitializes the log file when it reaches the maximum size.

Related reference

Environment variables for logging and tracing MQListener (Db2 Application programming and SQL)

Avoid external sorts for single-column column groups at the utility job level (PH03678 - February 2019)

When collecting statistics for a single-column column group, avoiding external data sorts can improve utility performance. Previously, you could control the sort behavior by using the STATCLGRST subsystem parameter. STATCLGRST specifies the amount of memory a utility can use for sorting for single-column groups instead of invoking an external sort program. However, STATCLGRST is a global setting, and if the value is large, excessive amounts of memory might be allocated for all utility jobs. Now, you can control this behavior at the utility level by using the new keyword STATCLGMEMSRT. This keyword applies to RUNSTATS, LOAD, and REORG TABLEPSACE.

Related reference

STATISTICS COLGROUP DATA SORT STG LIMIT field (STATCLGSRT subsystem parameter) (Db2 Installation and Migration)

LOAD (Db2 Utilities)
RUNSTATS (Db2 Utilities)
REORG TABLESPACE (Db2 Utilities)

User-defined function with MODIFIES SQL DATA from a subselect (PI93887 - January 2019)

This APAR introduces support in Db2 11 for invoking a user-defined function that is defined with MODIFIES SQL DATA in a subselect. Previously, Db2 issued SQLCODE -740 for this case.

The documentation of the MODIFIES SQL DATA option is also changed to include the following note:

Recommendation: If you invoke a function defined with the `MODIFIES SQL DATA` option in a subselect, make sure that the SQL statements inside the function do not modify any object that is referenced by that subselect.

Related reference
ALTER FUNCTION (compiled SQL scalar) (Db2 SQL)
ALTER FUNCTION (external) (Db2 SQL)
CREATE FUNCTION (compiled SQL scalar) (Db2 SQL)
CREATE FUNCTION (external scalar) (Db2 SQL)
Related information
-740 (Db2 Codes)

IBM zHyperLink: database synchronous read I/O exploitation (PH01123 - December 2018)

This APAR introduces support for performance improvements derived from IBM zHyperLink support. In a well-tuned environment, this feature can improve average response latencies for database page reads in online transactions.

The ZHYPERLINK subsystem parameter is introduced for enabling this feature, along with instrumentation to estimate the potential benefit of using the zHyperLink read support, and instrumentation to report the results of zHyperLink requests.

Related concepts
Read operations (Db2 Data Sharing Planning and Administration)
Related reference
DB2 zHyperLinks SCOPE field (ZHYPERLINK subsystem parameter) (Db2 Installation and Migration)

Allow ROTATE PARTITION for materialized query tables or tables having dependent materialized query tables (PH00194 - October 2018)

This APAR removes a restriction against rotating partitions when altering materialized query tables (MQTs) or tables with MQTs defined on them. Before this APAR, the ROTATE PARTITION clause of ALTER TABLE statements was restricted if the tables being altered were MQTs or tables with dependent MQTs.

Related tasks
Altering an existing materialized query table (Db2 Performance)
Related reference
ALTER TABLE (Db2 SQL)

Db2 command line processor (CLP) support for user name and credentials for running SQL files (PH00335 - September 2018)

This APAR introduces support for the Db2 command line processor to accept username and credential parameters when it is invoked to run SQL files. The credentials can be a password or RACF PassTicket and you can specify them in the new `-u` option flag when you start the Db2 command line processor.

Example: Specifying a user ID and password or RACF PassTicket to run an SQL file

Suppose that you want to use the Db2 command line processor to connect to a server and your system administrator provides the following connection URL for the server: `syszos1.abc.com:5021/ABCLOC1`

You can create a file named `script.sql` that contains the following statement:

```
CONNECT TO syszos1.abc.com:5021/ABCLOC1
```

Also suppose that the alias db2 has been set for the Db2 command line processor name, and your user ID is myid01. If your password is mypw01, you can use the following command to connect to the server:

```
db2 -f script.sql -u myid01/mypw01
```

If your credential is the RACF PassTicket phrase TMOK73SJ, you can use the following command to connect to the server:

```
db2 -f script.sql -u myid01/TMOK73SJ
```

Related reference

Start syntax for the Db2 command line processor (Db2 Commands)

CONNECT (Db2 command line processor) (Db2 Commands)

Related information

The RACF secured signon PassTicket

Suppress SQL warning messages for OPEN and FETCH from SPUFI, DSNTEP2, DSNTEP4, and DSNTIAUL (PI79053 - June 2018)

This APAR introduces the QUIET option for the TOLWARN parameter of SPUFI and the DSNTEP2, DSNTEP4, and DSNTIAUL programs. QUIET is the same as YES, except that the program suppresses all SQL warning messages from OPEN or FETCH statements if the SQLCODE is 0 or greater.

Related tasks

Controlling toleration of warnings in SPUFI (Db2 Application programming and SQL)

Related reference

DSNTEP2 and DSNTEP4 (Db2 Application programming and SQL)

DSNTIAUL (Db2 Application programming and SQL)

Profiles can start automatically when Db2 starts (PI89912 - January 2018)

In Db2 11, you can specify that profile take effect immediately when Db2 starts. The PROFILE_AUTOSTART subsystem parameter specifies whether a START PROFILE command is issued automatically when Db2 is started.

Related tasks

Monitoring and controlling Db2 with profile tables (Db2 Administration Guide)

Related reference

PROFILE AUTOSTART field (PROFILE_AUTOSTART subsystem parameter) (Db2 Installation and Migration)

-START PROFILE (Db2) (Db2 Commands)

Automatic remigration binds are removed in release coexistence and fallback (PI87675 - January 2018)

This APAR removes the possibility of disruptive repeating rebinds during release coexistence and fallback and remigration scenarios. The ABIND subsystem parameter is changed so that COEXIST behavior is used if YES is specified.

During release coexistence, or during fallback and re-migration, running a plan or package for the first time on the lower Db2 release results in an automatic bind. However, no automatic bind is triggered when a plan or package first runs again on the higher Db2 release.

When you are certain the Db2 is stable on the higher release and coexistence is no longer and issue, you can explicitly rebind the packages on the higher release.

Related concepts

Automatic binds in coexistence (Db2 Installation and Migration)

Related reference

AUTO BIND field (ABIND subsystem parameter) (Db2 Installation and Migration)

DSNTIJSG installation job no longer creates SYSIBM EXPLAIN tables (PI86450 - December 2017)

This APAR changes the DSNTIJSG installation job to remove CREATE statements for EXPLAIN tables with the SYSIBM qualifier. Db2 no longer uses these tables.

Related tasks

Creating EXPLAIN tables (Db2 Performance)

Related reference

EXPLAIN tables (Db2 Performance)

MQListener automatic restart after Db2 restart (PI84698 - December 2017)

This APAR introduces automatic restart of MQListener processing after Db2 is stopped and restarted, with the **-reconnectDB2** option of the **db2mqln1** and **db2mqln2 run** commands. Before this enhancement, you must restart the MQlistener after stopping and restarting Db2, which can cause service interruptions.

If you specify `-reconnectDB2 Y`, MQListener automatically reconnects and resumes processing after Db2 is stopped and restarted.

You can specify `-reconnectDB2 N` to continue with the previous behavior.

Related tasks

Configuring MQListener tasks (Db2 Application programming and SQL)

LOAD RESUME SUPPORT FOR INLINE COPY (PI81723 - December 2017)

To avoid unavailability of the table space or table space partition, the LOAD syntax has been enhanced to allow the COPYDDN or RECOVERYDDN option to be specified when RESUME YES is specified. A full image copy is taken at the end of LOAD processing, after potential BACKOUT processing.

Related reference

Syntax and options of the LOAD control statement (Db2 Utilities)

Hybrid Transactional Analytical Processing support for accelerated queries (PI83286 and PI83288 - November 2017)

APARs PI83286 and PI83288, which are packaged by a single PTF, UI51280, introduce Db2 for z/OS support for Hybrid Transactional Analytical Processing (HTAP) of accelerated dynamic queries.

With this new functionality, accelerated analytical dynamic queries have access to the latest transactional data that has been committed in Db2. Db2 analytical dynamic queries that are run on the analytics accelerator can now return results based on real-time transactional data with no data replication latency, so both your transactional and analytical queries run on a single system where it seems that real-time analytical processing on real-time data occurs even for queries that run on an accelerator. Data replication latency now no longer impacts Db2 dynamic query result consistency when using an analytics accelerator.

This HTAP functionality involves Db2 11 for z/OS as the database for the real-time transactional data, IBM Db2 Analytics Accelerator for z/OS Version 5.1 as the data repository for the analytical data, and IBM InfoSphere® Data Replication Change Data Capture for z/OS V10.2.1 to replicate the real-time transactional data from the Db2 database to the analytics accelerator. Additional PTFs are required for IBM Db2 Analytics Accelerator for z/OS and for IBM InfoSphere Data Replication Change Data Capture for z/OS.

Related reference

CURRENT QUERY ACCELERATION WAITFORDATA (Db2 SQL)

SET CURRENT QUERY ACCELERATION WAITFORDATA (Db2 SQL)

Related information

Making queries wait for incremental updates

z/OS DFSMS data set encryption (PI81900 - November 2017)

This APAR introduces enhancements that provide a simple, transparent, and consumable approach to enabling pervasive encryption of data at rest.

To implement the new encryption features, your security or storage administrator enables z/OS DFSMS data set encryption on your Db2 data sets. z/OS DFSMS data set encryption is a new hardware and software solution that is introduced in z/OS V2R3, and is also available through z/OS V2R2 APARs.

z/OS DFSMS data set encryption uses a key label to encrypt and decrypt the data. The key label is a string from 1 to 64 bytes that identifies a protected data key in the ICSF key repository.

You can use z/OS DFSMS data set encryption to protect all of your Db2 system-managed and user-managed objects, such as Db2-managed table space and index space data sets, data sets that are used by Db2 utilities, and sequential input and output data sets.

Related concepts

Encrypting your data with z/OS DFSMS data set encryption (Managing Security)

Support for collection of more accurate real storage usage statistics (PI82191 - August 2017)

PSPI

IFCID 0225 records contain new fields for collection of statistics about the number of discarded pages that are eligible for page steal. The IFCID 0225 record that is written at the end of the first one-minute statistics interval after midnight local time contains those statistics if the following conditions are true:

- Subsystem parameter REALSTORAGE_MANAGEMENT is set to ON or AUTO.
- A trace for IFCID 0503 is started.
- There are discarded frames.
- The PTF for z/OS APAR OA50366 is applied.

You can use those statistics in conjunction with existing real storage usage fields to calculate accurate storage usage once each day. By examining daily storage statistics for multiple weeks, you can determine whether there is an unusual increase in real storage usage.

See the description of IFCID 0225 in *prefix*.SDSNIVPD(DSNWMSGS) for details.

Important: You might see a performance impact during the time that the new statistics are collected. To avoid unexpected side effects, you might want to enable this enhancement in collaboration with IBM Support.

PSPI

Explicitly-defined hidden ROWID columns (PI77719 - June 2017)

This APAR removes a restriction against explicitly defining a ROWID column with the IMPLICITLY HIDDEN attribute. Columns with the IMPLICITLY HIDDEN attribute are not returned for statements that do not

explicitly list the column name in the select-clause. For example, the result for SELECT * does not include any implicitly hidden columns. To be included in the result, implicitly hidden columns must be explicitly specified in the select list.

Related concepts

Row ID values (Db2 SQL)

ROWID data type (Introduction to Db2 for z/OS)

New LOAD utility IGNORE options (PI79716 - June 2017)

This APAR introduces support for the following new IGNORE options for the LOAD utility.

PART
Specifies that records that do not satisfy any partition being loaded are ignored.

CONV
Specifies that records that cause a conversion error are ignored.

VALPROC
Specifies that records that fail a validation procedure are ignored.

IDERROR
Specifies that records that have an identity column value that is out of range are ignored.

DUPKEY
Specifies that records that cause a duplicate key error are ignored.

Related reference

Syntax and options of the LOAD control statement (Db2 Utilities)

ADMIN_INFO_SQL stored procedure enhanced to collect UDF information (PI45103 - April 2017)

This APAR enhances the ADMIN_INFO_SQL stored procedure and the DSNADMSB program to support the collection of information about user-defined functions (UDFs).

Related concepts

User-defined functions (Db2 SQL)

Related reference

ADMIN_INFO_SQL stored procedure (Db2 SQL)

DSNADMSB (Db2 Utilities)

LOAD utility support for more date and time formats (PI68139 - November 2016)

With this APAR the LOAD utility supports a wider range of data and time formats. It introduces the following new DATE EXTERNAL (*date-format*) and TIME EXTERNAL (*time-format*) specifications.

(*date-format*)
The format of the date representation, as shown in the following table.

Table 1. Values for date-format

***date-format* value**	**Format**	**Length**
DATE_A	*mm-dd-yyyy* "1" on page 10	10 bytes
DATE_B	*mm-dd-yy* "1" on page 10, "2" on page 10	8 bytes
DATE_C	*yyyy-mm-dd* "1" on page 10	10 bytes
DATE_D	*yy-mm-dd* "1" on page 10, "2" on page 10	8 bytes
DATE_E	*dd-mm-yyyy* "1" on page 10	10 bytes

Table 1. Values for date-format (continued)

***date-format* value**	**Format**	**Length**
DATE_F	*dd-mm-yy* "1" on page 10, "2" on page 10	8 bytes
DATE_G	*yyyy-ddd* "1" on page 10	8 bytes
DATE_H	*yy-ddd* "1" on page 10, "2" on page 10	6 bytes
DATE_I	*mmddyyyy*	8 bytes
DATE_J	*mmddyy* "2" on page 10	6 bytes
DATE_K	*yyyymmdd*	8 bytes
DATE_L	*yymmdd* "2" on page 10	6 bytes
DATE_M	*ddmmyyyy*	8 bytes
DATE_N	*ddmmyy* "2" on page 10	6 bytes
DATE_O	*yyyyddd*	7 bytes
DATE_P	*yyddd* "2" on page 10	5 bytes

If a date format is specified for a field that is used in a field specification, the field specification must also use the specified date format.

Notes:

1. If the format includes separators, the separator can be any single-byte character that can be converted to a single-byte EBCDIC character. However, if the input file has a delimited format, you cannot specify the same character that is used for a delimiter, including the COLDEL, CHARDEL, and DECPT delimiters. For more information about delimiter restrictions, see Loading delimited files (Db2 Utilities).
2. If a two-digit year is specified, it is expanded to a four-digit year. If the two-digit year is less than the sum of the two-digit current year plus 50, then the current century is used in the four-digit year. For example, assume that the current year is 2017. If the two-digit year is 67, 1967 is used. However, if the two-digit year is 66, 2066 is used.

(*time-format*)
The specific format of the time representation, as shown in the following table.

Table 2. Values for time-format

***time-format* value**	**Format**	**Length**
TIME_A	*hh.mm.ss* "1" on page 11	8 bytes
TIME_B	*hh.mm* "1" on page 11	5 bytes
TIME_C	*hh.mm* AM or *hh.mm* PM "1" on page 11	8 bytes
TIME_D	*hhmmss*	6 bytes
TIME_E	*hhmm*	4 bytes

If a time format is specified for a field that is used in a field specification, the field specification must also use the specified time format.

Note:

1. If the format includes separators, the separator can be any single-byte character that can be converted to a single-byte EBCDIC character.

Related reference

Syntax and options of the LOAD control statement (Db2 Utilities)

Reduced cost for collecting Db2 frequency statistics (PI74408 - August 2017)

Your enterprise now has more control over performance when you collect frequency statistics for single-column column groups through RUNSTATS or inline statistics jobs. By adjusting the value of a new subsystem parameter, STATCLGSRT, you can potentially improve performance and reduce costs.

The STATCLGSRT subsystem parameter specifies the amount of memory that Db2 can use to avoid a sort operation for RUNSTATS or other inline statistics utility jobs when FREQVAL is specified on a COLGROUP that identifies one or more single-column column groups. By increasing the value of STATCLGSRT to allocate more memory, you enable Db2 to collect the statistics you need without performing a costly sort.

Related tasks

Reducing the cost of collecting statistics (Db2 Performance)

Related reference

STATISTICS COLGROUP DATA SORT STG LIMIT field (STATCLGSRT subsystem parameter) (Db2 Installation and Migration)

Syntax and options of the LOAD control statement (Db2 Utilities)

Syntax and options of the REORG TABLESPACE control statement (Db2 Utilities)

RUNSTATS TABLESPACE syntax and options (Db2 Utilities)

LOAD REPLACE SHRLEVEL REFERENCE (PI67793 - November 2016)

This APAR introduces support for LOAD REPLACE SHRLEVEL REFERENCE, which enabled concurrent read application access to the target table while new data is being loaded into a set of shadow data sets. A new SWITCH phase is introduced to the LOAD utility to switch access between the original and shadow data sets.

Related reference

Syntax and options of the LOAD control statement (Db2 Utilities)

Chapter 2. Performance

In Db2 11 of Db2 for z/OS, performance improvements focus on optimizing query processing and reducing CPU processing time without causing significant administration or application changes.

This release of Db2 for z/OS runs on only IBM System z10® and subsequent 64-bit z/Architecture® processors, using z/OS Version 1 Release 13 or later operating systems. This configuration reduces CPU consumption from the start.

Related information

Db2 11 for z/OS Performance Topics (IBM Redbooks)

Subsystem and Transaction Monitoring and Tuning with Db2 11 for z/OS (IBM Redbooks)

Db2 11 for z/OS: Buffer Pool Monitoring and Tuning (IBM Redbooks)

Support for indexes that exclude NULL keys

In Db2 11, you can improve the performance of an index by excluding NULL keys from an index to reduce its size.

NULL key columns add to the size of an index and can reduce the performance of index scans. If you exclude NULL key columns from an index, Db2 creates index entries only for key columns that are not null. You can specify that an index excludes NULL keys when you create an index by using the CREATE INDEX statement.

Related concepts

Indexes that exclude NULL keys (Introduction to Db2 for z/OS)

Improved performance of stored procedure calls from ODBC applications

In Db2 11, Db2 improves the processing of stored procedure calls from a local ODBC application.

Db2 optimizes the communication between the Db2 ODBC driver and Db2 to execute CALL statements and return result sets more efficiently. In addition, Db2 improves the processing of the result sets that are returned by using limited block fetch and progressive streaming.

Related concepts

ODBC limited block fetch (Db2 Programming for ODBC)

Improved performance of ODBC applications that retrieve LOB or XML objects
For ODBC applications, Db2 11 uses limited block fetch by default to retrieve small inline and large LOBs or XML objects. This new behavior can improve performance.

Improved performance of ODBC applications that retrieve LOB or XML objects

For ODBC applications, Db2 11 uses limited block fetch by default to retrieve small inline and large LOBs or XML objects. This new behavior can improve performance.

When limited block fetch is enabled, you can control whether LOB and XML objects are returned inline by using the new STREAMBUFFERSIZE initialization keyword.

To disable limited block fetch, set the LIMITEDBLOCKFETCH initialization keyword to 0.

Related concepts

ODBC limited block fetch (Db2 Programming for ODBC)

Improved performance of stored procedure calls from ODBC applications
In Db2 11, Db2 improves the processing of stored procedure calls from a local ODBC application.

Related reference

Db2 ODBC initialization keywords (Db2 Programming for ODBC)

© Copyright IBM Corp. 2013, 2019

Buffer pool enhancements

In Db2 11, several buffer pool enhancements help to improve performance by controlling the size of a buffer pool and helping you estimate the appropriate size for the buffer pool.

- Now, the Db2 command ALTER BUFFERPOOL provides the keywords VPSIZEMIN and VPSIZEMAX. You can use these keywords to specify the minimum and maximum number of buffer pools to allocate to the active buffer pool when AUTOSIZE(YES) is in effect.
- You can specify the frame size for a buffer pool by using the new keyword FRAMESIZE.

  PGFIX and FRAMESIZE at migration: Buffer pools that are set to PGFIX(YES) during migration to Db2 11 can use FRAMESIZE(1M) by default in Db2 11. However, buffer pools that are set to PGFIX(NO) during migration use 4 KB frames in Db2 11 even if altered to PGFIX(YES), unless you explicitly specify a different FRAMESIZE value.
- You can do buffer pool simulations while you run your regular workloads to determine how the buffer pool size affects the number of synchronous read I/O operations. The ALTER BUFFERPOOL command with the SPSIZE and SPSEQT options starts a buffer pool simulation.

Related reference

-ALTER BUFFERPOOL (Db2) (Db2 Commands)

Performance improvements for the LOAD utility

In Db2 11, the elapsed time of some LOAD utility jobs is reduced because LOAD can use parallelism in more situations than in earlier versions of Db2. LOAD can use parallelism for partitioned table spaces with only one input data set. Also, LOAD can use parallelism for non-partitioned table spaces.

In earlier versions of Db2, LOAD uses parallelism for partitioned table spaces only if you provide a separate input data set for each partition. Providing and maintaining such data sets is difficult, especially if you altered the partition boundaries and then had to redistribute the data across the input data sets. Also, LOAD cannot use parallelism for simple or segmented table spaces.

In Db2 11, these restrictions are removed. You no longer need to provide separate input data sets for each partition so that LOAD can use parallelism. Also, LOAD can use parallelism for non-partitioned table spaces. Specifically, LOAD can use parallelism with one input data set for all of the following table space types:

- Simple table spaces
- Segmented (non-UTS) table spaces
- Partitioned (non-UTS) table spaces
- partition-by-range table spaces

As a result, the elapsed time for the load operation can be reduced.

To specify that you want LOAD to use parallelism if possible, use the new PARALLEL keyword in the LOAD utility control statement.

Related concepts

Table space types and characteristics in Db2 for z/OS (Db2 Administration Guide)

Related tasks

Improving LOAD performance (Db2 Utilities)

Related reference

Syntax and options of the LOAD control statement (Db2 Utilities)

Performance improvements for the REORG TABLESPACE utility

Db2 11 delivers improved performance for the REORG TABLESPACE utility with the PART option.

In Db2 11, when subsystem parameter REORG_PART_SORT_NPSI is enabled, REORG TABLESPACE PART sorts all keys of the nonpartitioned secondary indexes and builds the shadow index from the sorted keys. This improvement potentially reduces CPU time.

In addition, this version delivers potential performance improvements as the following items increase:

- The number of nonpartitioned secondary indexes on the table space
- The amount of data in the logical part of the nonpartitioned secondary indexes
- The total amount of data in the nonpartitioned secondary indexes

Related reference

REORG TABLESPACE (Db2 Utilities)

REORG PART SORT NPSI field (REORG_PART_SORT_NPSI subsystem parameter) (Db2 Installation and Migration)

Improvements for RELEASE(DEALLOCATE) processing

Db2 11 offers improvements that make the RELEASE(DEALLOCATE) bind option the best choice in more situations.

The RELEASE(DEALLOCATE) bind option causes resources for Db2 packages to be freed when the threads that are associated with the packages are deallocated. This behavior provides an opportunity for better performance for many applications. However, in previous releases of Db2, a limitation of RELEASE(DEALLOCATE) behavior was that it was not possible for data definition statements or bind operations to break in to long-running applications. The following improvements in Db2 11 overcome this limitation.

Break-in for active threads

The new PKGREL_COMMIT subsystem parameter specifies whether, at COMMIT or ROLLBACK, a persistent Db2 thread releases a package that is active on that thread if certain Db2 operations are waiting for exclusive access to that package. If you specify YES, for packages that are bound with the RELEASE(DEALLOCATE) option, the following operations are permitted at COMMIT or ROLLBACK while the package is active and allocated by Db2 for a persistent Db2 thread:

- BIND PACKAGE with REPLACE and REBIND PACKAGE requests, including automatic rebind online schema changes for tables and indexes that are statically referenced by the package
- Online REORG operations that materialize pending definition changes for objects that are statically referenced by the package

With this improvement, you no longer must identify and stop persistent Db2 threads that are running packages that are bound with the RELEASE(DEALLOCATE) bind option before you attempt BIND PACKAGE with REPLACE and REBIND PACKAGE requests, or apply schema changes that are associated with those packages.

Break-in for idle threads

If a package lock is likely to time out, Db2 recycles any idle threads that use local attachment facilities, such as CICS, IMS, or RRSAF. A thread is eligible for recycling if it is has these characteristics:

- Is at a transaction boundary
- Is not running in Db2
- Has not committed or rolled back recently

If an application attempts to use an idle thread during the recycle processing, use of the thread is delayed until the recycle process is complete.

Related reference

RELEASE bind option (Db2 Commands)

Faster recovery of catalog and directory objects

In Db2 11, the recovery time of certain catalog and directory objects can be faster because unnecessary log scans are reduced.

The following directory objects and any associated indexes are now recorded in DSNDB01.SYSLGRNX:

- DSNDB01.SCT02
- DSNDB01.SPT01
- DSNDB01.SYSSPUXA
- DSNDB01.SYSSPUXB

When one of these objects is open for updates, the SYSLGRNX table stores the log range for the update. When these objects must be recovered, Db2 can use the SYSLGRNX records to determine the appropriate log records to read.

Before Db2 11, when these directory objects were not recorded in SYSLGRNX, Db2 could not determine which log records to read and potentially had to scan unnecessary log data. This extra scanning could significantly increase the recovery time of these objects. This increased recovery time also delayed the recovery of the other objects that were included in the same recovery step. (As part of the procedure to recover the catalog and directory, these directory objects are grouped with other objects in the same RECOVER statement.)

This enhancement can improve the recovery time of not only these four directory objects that are now recorded in SYSLGRNX, but of all catalog and directory objects that are recovered in the same recovery step.

The recording of these objects in SYSLGRNX is supported in Db2 11 conversion mode. Db2 can use these SYSLGRNX records for recovery in Db2 11 new-function mode.

Related concepts

Point-in-time recovery of the catalog, directory, and all user objects (Db2 Utilities)

Related tasks

Recovering catalog and directory objects (Db2 Utilities)

Reduction in CPU usage for large numbers of partitions

Db2 11 offers performance improvements by reducing CPU usage when table spaces that have a large number of partitions are accessed.

In earlier versions of Db2 for z/OS, you might experience performance degradation after greatly increasing the number of partitions in a table space. Db2 11 optimizes partition processing so that CPU usage does not increase dramatically as the number of partitions increase.

CPU reduction for distributed applications

Db2 11 reduces CPU usage for distributed applications that select many columns. This enhanced performance is a result of the improved way that Db2 processes the columns.

Improved performance for ROLLBACK TO SAVEPOINT

Db2 11 provides improved performance for the ROLLBACK TO SAVEPOINT statement.

In earlier versions of Db2, the performance of this statement progressively deteriorates with repeated rollbacks to the same external savepoint. This performance degradation is caused by the increasing number of log records that are being scanned each time that the ROLLBACK TO SAVEPOINT statement is issued.

In Db2 11, the ROLLBACK TO SAVEPOINT statement recognizes the point in the log where the previous savepoint finished. This enhancement improves performance by ensuring that subsequent executions of

the ROLLBACK TO SAVEPOINT statement scan only the log records that apply to the amount of work to be undone.

Db2 DPSI performance enhancements

In Db2 11, you can specify the maximum degree of parallelism for a parallel group in which a data partitioned secondary index (DPSI) is used to drive parallelism.

A DPSI is a nonpartitioning index that is physically partitioned according to the partitioning scheme of the table. When you specify a value of greater than 0 for the new PARAMDEG_DPSI subsystem parameter, you limit the degree of parallelism for DPSIs. When you limit the degree of parallelism, Db2 does not create extraneous parallel tasks that use virtual storage.

Reductions in synchronous log writes

Db2 11 improves performance and stability by reducing the occurrence of synchronous log writes during index structure modification operations in data sharing environments.

Previously, performance problems might result during insert and delete processing for group buffer pool dependent (GBP-dependent) indexes in data sharing environments. Synchronous log writes for the same operation might cause performance degradation.

Now, the number of synchronous log writes for both insert and delete processing is reduced, which improves transaction and batch response times.

Relief from storage shortages in group buffer pools

In Db2 11, data sharing members can automatically respond to storage shortages in group buffer pools with minimal impact to your applications. This new behavior is called the *group-buffer-pool write-around protocol*.

Before Db2 11, when a group buffer pool sustained a large amount of page-write activity, the coupling facility for the group buffer pool could quickly fill up with changed pages. As a result, applications could be slower, and in severe cases, pages could be placed in the LPL. These LPL pages could result in Db2 data outages.

In Db2 11, Db2 automatically detects large amounts of page-write activity and switches to the group-buffer-pool write-around protocol for those objects that have the most write activity. (Typically, the objects that have the most write activity are utility and batch jobs.) Under this new protocol, pages that are changed by write operations are written to the group buffer pool only if they are already cached there. Otherwise, Db2 writes these pages to disk. The other objects (objects that are using the group buffer pool but do not have the most write activity) are not affected by this new protocol. They are able to access the group buffer pool as usual. When the storage shortage is relieved, Db2 switches back to normal write activity for the group buffer pool.

Db2 automatically implements the group-buffer-pool write-around protocol when necessary. However, the following prerequisite software is required:

- z/OS Version 1 Release 12 or Release 13 with the PTFS for OA37550 or z/OS Version 2 Release 1
- Coupling facility control code (CFCC) at CFLEVEL=17

To determine how many pages are written to disk by the group-buffer-pool write-around protocol, issue the DISPLAY GROUPBUFFERPOOL command with the MDETAIL option. In the output from this command, look at the WRITE-AROUND PAGES field in message DSNB777I.

Related reference

-DISPLAY GROUPBUFFERPOOL (Db2) (Db2 Commands)

Automatic cleanup of pseudo-empty index pages

Db2 11 automatically cleans up pseudo-deleted index entries and pseudo-empty index pages. This enhancement can reduce the size of some indexes, which can improve SQL performance and reduce the need to run the REORG INDEX utility.

When rows are deleted, index entries are not physically deleted unless the delete operation has exclusive control over the index page set. These index entries are called *pseudo-deleted index entries*. Subsequent searches continue to access these pseudo-deleted entries, which can gradually degrade performance as more rows are deleted. These pseudo-deleted index entries can also result in timeouts and deadlocks for applications that insert data into tables with unique indexes.

Before Db2 11, Db2 attempted to clean up pseudo-empty index pages as part of SQL DELETE processing. *Pseudo-empty index pages* are pages that contain only pseudo-deleted index entries. However, if some of the pseudo-deleted entries in the page were not committed during the SQL DELETE processing, cleanup could not be performed. Therefore, some pseudo-empty pages were likely not cleaned up.

In addition to the cleanup that was previously done, Db2 11 also automatically deletes pseudo-empty index pages independently of the SQL DELETE transaction. Db2 also automatically deletes individual pseudo-deleted index entries.

Automated cleanup of pseudo-empty index pages and pseudo-deleted index entries is enabled for all indexes by default when the value of the INDEX_CLEANUP_THREADS subsystem parameter is set to a non-zero value. However, you can specify time windows to enable or disable the index cleanup for the entire subsystem, for indexes in specific databases, or for specific indexes.

When the system is zIIP-enabled, this automated clean-up function runs under enclave service request blocks (SRBs) that are zIIP-enabled. However, the system does not need to be zIIP-enabled to use this new function.

This new function is designed to run with minimal or no disruption to other concurrent Db2 work in the system.

Related tasks

Configuring IBM Z Integrated Information Processor (zIIP) specialty engines for Db2 (Db2 Performance)

Controlling index cleanup processing (Db2 Performance)

Improving concurrency (Db2 Performance)

Related reference

INDEX CLEANUP THREADS field (INDEX_CLEANUP_THREADS subsystem parameter) (Db2 Installation and Migration)

Improved duplicate removal from SQL results

In Db2 11, Db2 improves the processing of removing duplicates from SQL results.

Db2 might optimize the processing of certain types of SQL statements that remove duplicates during processing. Some improvements require an index. Examples of SQL statements that are affected include:

- SELECT DISTINCT
- Single SET FUNCTION DISTINCT
- GROUP BY without set functions
- GROUP BY with single SET FUNCTION DISTINCT
- Single MAX or MIN set function
- GROUP BY with single MAX or MIN set function
- SELECT with a subquery

Performance improvements for distributed applications

Db2 11 provides improved performance for distributed applications that return large result sets.

Db2 11 introduces package-based continuous block fetch, which can improve performance for retrieval of large, read-only result sets from a remote Db2 for z/OS server.

Like SQL-based continuous block fetch, package-based continuous block fetch causes fewer messages to be transmitted from the requester to retrieve the entire result set. However, package-based continuous block fetch is easier to configure. It requires only that you bind your applications with the new DBPROTOCOL(DRDACBF) option. You do not need to modify your applications or set subsystem parameters to indicate the maximum number of blocks to be returned for a remote request.

In addition, package-based continuous block fetch is more efficient than SQL-based continuous block fetch. With package-based continuous block fetch, the requester opens a secondary connection to the Db2 server for each read-only cursor. The Db2 server returns extra query blocks until all of the rows for the cursor are retrieved. When the cursor is closed, the secondary connection is implicitly closed.

Package-based continuous block fetch provides a performance advantage for a Db2 for z/OS application that has the following characteristics:

- The application queries only remote sites.
- The application does not contain INSERT, UPDATE, DELETE, or MERGE statements.
- No statement in the application creates a unit of recovery on the remote site. This situation results in an SQL error when the application package is bound for package-based continuous block fetch.

Expanded RBA and LRSN log records

In Db2 11, the RBA and LRSN log records are expanded from basic 6-byte format to extended 10-byte format.

In earlier versions, if you reach the end of the RBA log range, you must manually reset the RBA back to zero.

Beginning in Db2 11, you can convert the RBA and LRSN to extended 10-byte format to avoid reaching the logging limits. Also, new warning messages alert you when the LRSN is approaching the end of its range.

In Db2 11, the extended RBA and LRSN values might be externalized before objects are converted in certain contexts.

Related concepts

The extended 10-byte RBA and LRSN in Db2 11
Beginning in Db2 11, Db2 uses 10-byte RBA and LRSN values.

How RBA and LRSN values are displayed (Db2 Administration Guide)

Related tasks

What to do before RBA or LRSN limits are reached (Db2 Administration Guide)

Converting page sets to the 10-byte RBA or LRSN format (Db2 Administration Guide)

Related reference

Utility RBA and LRSN handling in Db2 11

Related information

Db2 11 for z/OS Technical Overview (IBM Redbooks)

Improved access path reuse with warnings

The new APREUSE(WARN) bind option improves the ability to reuse existing access paths for as many statements as possible.

In earlier versions, access path stabilization through the APREUSE option of the BIND PACKAGE and REBIND PACKAGE commands is possible only when the access paths for all statements in a package can

be reused successfully. If reuse fails for any statement in a package, reuse cannot be applied to any statements in that package.

In Db2 11, the APREUSE(WARN) option enables a package to be successfully rebound when the access paths for some statements cannot be reused. When an access path cannot be reused, a new access path is chosen for the statement.

Related tasks

Reusing and comparing access paths at bind and rebind (Db2 Performance)

Related reference

APREUSE bind option (Db2 Commands)

Free space management enhancements

In Db2 11, you can reserve free space in table spaces for use only by UPDATE operations.

By reserving the space for UPDATE operations, you can improve performance. The free space enables your data to remain clustered longer and reduces the use of overflow records and indirect references, which degrade performance.

You can specify the PCTFREE FOR UPDATE option when you create or alter a table space. When you specify that option, Db2 reserves space for future UPDATE operations. The reserved space is not used by INSERT operations or utilities that add data to the table space. The FOR UPDATE value specifies the percentage of each page that is reserved to be used only by future UPDATE operations. You can specify `FOR UPDATE -1` to specify that Db2 uses real-time statistics to automatically calculate how much free space to reserve for updates.

The PCTFREE_UPD subsystem parameter specifies the default value of the PCTFREE FOR UPDATE option.

Related tasks

Reserving free space for table spaces (Db2 Performance)

Related reference

ALTER TABLESPACE (Db2 SQL)

CREATE TABLESPACE (Db2 SQL)

PERCENT FREE FOR UPDATE field (PCTFREE_UPD subsystem parameter) (Db2 Installation and Migration)

Improved predicate transformations

Db2 11 applies more techniques to transform query predicates for processing at earlier stages.

Db2 applies the following types of query transformations to make predicates eligible to be processed at earlier stages:

- Addition of generated predicates
- Removal of unneeded pre-evaluated predicates
- Removal of unneeded table references in certain join predicates

Generally, you do not have to do anything to take advantage of the new query transformations.

Related concepts

Predicate manipulation (Db2 Performance)

Query transformations (Db2 Performance)

Predicates that Db2 generates (Db2 Performance)

Stage 1 and stage 2 predicates (Db2 Performance)

Indexable and non-indexable predicates (Db2 Performance)

Related reference

Summary of predicate processing (Db2 Performance)

Predicate selectivity overrides

You can provide information to Db2 about predicates that are difficult for Db2 to estimate. Db2 uses the information that you provide during access path selection.

Db2 cannot estimate filter factors for certain types of predicates. It must rely on inaccurate estimates or even default filter factors to select access paths for statements that use such predicates. Examples include predicates that involve the following items, among others:

- Host variables
- Parameter markers
- Expressions
- Table self-joins
- Subqueries

However, you can override these default filter factors for certain predicates by creating selectivity overrides. Each predicate selectivity override describes the selectivity of a particular predicate in a particular SQL statement. When a statement contains more than one predicate, you can create separate selectivity overrides for each predicate in the statement. To create selectivity overrides, you populate a set of input EXPLAIN tables and issue a BIND QUERY command.

The selectivity overrides are added to certain catalog tables, and Db2 uses the information during access path selection.

Related tasks

Overriding predicate selectivities at the statement level (Db2 Performance)

Managing query access paths (Db2 Performance)

Related reference

BIND QUERY (DSN) (Db2 Commands)

DSN_PREDICAT_TABLE (Db2 Performance)

DSN_PREDICATE_SELECTIVITY table (Db2 Performance)

DSN_USERQUERY_TABLE (Db2 Performance)

Chapter 3. Availability

Db2 11 continues to offer improvements in availability, such as support for altering the limit keys for a partitioned table space without impacting the availability of the data, and support for online REORG and work file database enhancements.

Automatic recovery of indexes from GRECP or LPL status

Db2 11 improves system and application availability by extending support for the automatic recovery of indexes that are in group buffer pool recovery pending (GRECP) or logical page list (LPL) status during GRECP or LPL recovery.

In earlier versions of Db2, if an index page split or index page delete operation is unfinished when an index is put into GRECP or LPL status, the GRECP or LPL recovery might fail, and the index can be left in REBUILD-pending status. In these cases, a database administrator must rebuild the index, which is a time-consuming process.

Now, Db2 automatically uses second pass log apply to initiate recovery of an index that is in GRECP or LPL status. This enhancement makes the indexes immediately available after GRECP or LPL recovery.

In some rare cases when you run the RECOVER utility or RESTORE SYSTEM utility, an index might be left in REBUILD-pending status. In these cases, you must rebuild the index by running the REBUILD INDEX utility.

Related concepts

Recovery of pages on the logical page list (Db2 Data Sharing Planning and Administration)

Effects of running RECOVER (Db2 Utilities)

After running RESTORE SYSTEM (Db2 Utilities)

Related information

DSNB317I (Db2 Messages)

DSNB328I (Db2 Messages)

Online schema enhancements

Db2 11 improves system and application availability by allowing recovery to a point in time before materialization of pending definition changes.

DB2 10 improved the availability of table spaces and indexes by allowing *pending definition changes*. Pending definition changes are changes that are made by ALTER statements, but are not yet materialized. Examples are changing segment size, data set size, buffer pool page size, and the MEMBER CLUSTER attribute. Users can defer materialization of the changes until a convenient time. Materialization is accomplished by running the REORG utility. However, in DB2 10, after the REORG utility was run on the objects, you could not recover those objects to a point in time before the changes were materialized.

In Db2 11 you can recover the following objects to a point in time before the pending definition changes were materialized:

- partition-by-range table spaces
- LOB or XML table spaces

The pending definition changes must meet the following criteria:

- The changes were materialized in Db2 11 new-function mode.
- The changes were not changes to the table space type or the hash organization.

Related concepts

Point-in-time recovery (Db2 Utilities)

© Copyright IBM Corp. 2013, 2019

Improved data availability when altering limit keys

In Db2 11, you can alter the limit keys for partitioned table spaces without impacting the availability of the data.

In earlier versions of Db2, when you change limit key values, all affected partitions are placed in REORG-pending (REORP) status. The data becomes unavailable until you run the REORG utility.

In Db2 11, that restriction is removed. When you change the limit key values, the data remains available and applications can continue to access the data. In most cases however, the limit key changes are not materialized until the next time that you run REORG. The affected partitions are placed in advisory REORG-pending (AREOR) status. This type of definition change, in which the changes are not immediately materialized, is called a *pending data definition change*. For more information about such changes, see Pending data definition changes (Db2 Administration Guide).

Important: The introduction of pending changes for limit keys affects how you materialize changes to limit keys. For more information about changes that might be required in your Db2 environment, see "ALTER statements that change limit keys are pending changes" in “Application and SQL release incompatibilities” on page 47.

In a few cases, a change to a limit key value is immediately materialized, so you do not need to run REORG on the partitions. Immediate materialization occurs when Db2 determines that both of the following conditions are true:

- No data needs to be moved between partitions.
- No other alter limit key operation is pending on the partition.

This new capability applies to the following types of partitioned table spaces:

- partition-by-range table spaces
- Partitioned (non-UTS) table spaces with table-controlled partitioning

Beyond improved data availability, the ability to alter limit keys as pending changes provides other benefits:

- You can alter a limit key even when the table space has other pending definition changes.
- Because the materialization of the limit key changes is deferred, you have a time period where you can drop limit key changes without affecting the target table.

For partitioned (non-UTS) table spaces with index-controlled partitioning, altering the limit key still causes the affected partitions to be placed in REORG-pending (REORP) status. Therefore, the data is unavailable until the affected range of partitions is reorganized. To prevent such outages, you can use the new subsystem parameter PREVENT_ALTER_LIMITKEY. Specifying a value of YES for this parameter restricts ALTER TABLE statements that alter limit keys for index-controlled partitioned table spaces.

Related concepts

Ability to alter limit keys for materialized query tables
In Db2 11, you can alter limit keys for materialized query tables or tables that are referenced by materialized query tables.

Improvements to DROP PENDING CHANGES
You can remove pending table space definition changes that are not yet materialized by using the ALTER TABLESPACE statement with the DROP PENDING CHANGES clause. In Db2 11, this statement now resets advisory REORG-pending (AREOR) status.

Table space types and characteristics in Db2 for z/OS (Db2 Administration Guide)

Related tasks

Changing the boundary between partitions (Db2 Administration Guide)

Related reference

REORG-pending status (Db2 Utilities)

PREVENT ALTER LIMITKEY field (PREVENT_ALTERTB_LIMITKEY subsystem parameter) (Db2 Installation and Migration)

Automated REORG mapping table management

In Db2 11, you no longer have to create your own mapping table and index before running the REORG TABLESPACE utility. If you do not specify a mapping table in your utility statement, REORG creates one for you.

You can use the MAPPINGDATABASE keyword to specify the database in which REORG implicitly creates the mapping table and index. Alternatively, you can specify the default database to use for all mapping tables by using the REORG_MAPPING_DATABASE subsystem parameter.

Also, in Db2 11, the mapping table can be a partitioned-by-growth table space, which means that it can grow up to 16 TB. In previous versions, the mapping table was limited to 64 GB

You can still create your own mapping tables and indexes. However, be aware that the format of the mapping table has changed in Db2 11 to accommodate the new 10-byte format of RBA and LRSN values.

Related reference

REORG MAPPING DB field (REORG_MAPPING_DATABASE subsystem parameter) (Db2 Installation and Migration)

Syntax and options of the REORG TABLESPACE control statement (Db2 Utilities)

Work file database enhancements

Db2 11 provides improved monitoring of storage use in the work file database so that you can alleviate storage shortages before application failures occur.

In Db2 11, Db2 provides the following improvements to help you monitor disk storage use in the work file database:

- Two subsystem parameters that provide threshold levels for alerts about work file storage shortages, and two messages that provide the alerts:

  WFSTGUSE_AGENT_THRESHOLD
  This subsystem parameter enables you to control when a message is issued to indicate that an agent is using too much disk space in the work file database. WFSTGUSE_AGENT_THRESHOLD determines the percentage of available space in the work file database on a Db2 subsystem or data sharing member that can be consumed by a single agent before message DSNI052I is issued.

  WFSTGUSE_SYSTEM_THRESHOLD
  This subsystem parameter enables you to control when a message is issued to indicate that the total disk space that is being used in the work file database by all agents is too great. WFSTGUSE_SYSTEM_THRESHOLD determines the percentage of available space in the work file database on a Db2 subsystem or data sharing member that can be consumed by all agents before message DSNI053I is issued.

- New statistics class 1 trace records that track work file space usage.

Related reference

AGENT LEVEL THRESHOLD field (WFSTGUSE_AGENT_THRESHOLD subsystem parameter) (Db2 Installation and Migration)

SYSTEM LEVEL THRESHOLD field (WFSTGUSE_SYSTEM_THRESHOLD subsystem parameter) (Db2 Installation and Migration)

Governing of parallel processing of utilities

In Db2 11, you can manage the amount of parallelism by using the new PARAMDEG_UTIL subsystem parameter and the PARALLEL(*num-subtasks*) utility option.

When utilities use parallel processing, they consume more processor resources. This use of processor resources can be an issue for some utilities, such as CHECK INDEX, LOAD, REBUILD INDEX, REORG TABLESPACE, and UNLOAD. The new PARAMDEG_UTIL subsystem parameter and the PARALLEL(*num-subtasks*) utility option enable you to control the amount of parallelism by specifying the maximum number of subtasks that a utility can generate.

Improved availability of previous compression dictionaries

Db2 11 improves the availability of previous compression dictionaries by writing them to the log and making them available through the instrumentation facility interface (IFI) to data replication, log analysis, and similar tools and products.

With data compression, an initial compression dictionary is created during a LOAD, REORG, or SQL insert operation. In earlier versions of Db2, a REORG or LOAD REPLACE without KEEPDICTIONARY might occur and cause compression dictionaries older than two versions to be unavailable. In Db2 11, Db2 can efficiently access older compression dictionaries from the log through the instrumentation facility interface (IFI).

Related concepts

Contents of the log (Db2 Administration Guide)

Related reference

Qualifying log records (Db2 Administration Guide)

Db2 changes for the GDPS Continuous Availability with zero data loss solution

To support the GDPS Continuous Availability with zero data loss (GDPS Continuous Availability with zero data loss) solution, Db2 11 introduces subsystem parameters, commands, use of a new data set to support compressed log records, a new REORG TABLESPACE option, enhanced DISPLAY LOG command output, and new IFI qualification field values.

GDPS Continuous Availability with zero data loss solution terminology: A GDPS Continuous Availability with zero data loss solution that includes Db2 requires three data sharing environments. The solution includes a *source* Db2 data sharing group, a *proxy* Db2 data sharing group, and a *target* Db2 data sharing group. The proxy group uses a *capture* program to capture changes to tables on the source group. The proxy group then uses a *replication* program to replicate the changes to the target group. A VSAM key-sequenced data set, called the *compression dictionary data set (CDDS)*, must be defined in the source group. It holds the following items for use in capture and replication:

- The expansion dictionaries for Db2 tables whose changes are captured

  Currently, the CDDS contains a maximum of three versions of the expansion dictionaries.
- System status information that the proxy group uses to find the log data sets and to determine the status of the source group members

Subsystem parameters

The following subsystem parameters are added to support this configuration. These subsystem parameters cannot be changed online.

CDDS_MODE
Specifies whether a member of a Db2 data sharing group is part of a source data sharing group, a proxy data sharing group, or neither. Possible values are SOURCE, PROXY, and NONE.

CDDS_PREFIX
Specifies a prefix of up to 39 bytes for the CDDS name. The full CDDS name is this prefix, appended with '.CDDS'.

CDDS creation and population

You need to define the CDDS on the source data sharing group. You run the REORG TABLESPACE utility with the INITCDDS and SEARCHTIME options to populate the CDDS.

Commands for starting and stopping the CDDS

Before a CDDS can be recovered, it must be closed and deallocated. The -STOP CDDS command lets you close and deallocate a CDDS without stopping all members of a data sharing group. You can then recover

the CDDS, and issue the -START CDDS command to allocate and open the CDDS on all members of the data sharing group. You can issue the -START CDDS and -STOP CDDS commands from the source or proxy data sharing group.

DISPLAY LOG command output changes

When a data sharing group is enabled as a source or proxy group, the DISPLAY LOG command includes message DSNJ375I, which shows the name of the CDDS, and whether it is enabled in source or proxy mode.

DSNJU008 (print CDDS) utility

The DSNJU008 stand-alone utility prints the CDDS. You can filter the output by DBID, PSID, table space partition, or expansion dictionary version.

IFI qualification changes

PSPI

The IFI READS command for IFCID 0306 supports capture of log records. The following values are added to the WQALLCRI field to indicate that log records are being collected by the proxy data sharing group for the source data sharing group.

X'01' (WQALLCR1)
Only log records for changed data capture and unit of recovery control from the proxy data sharing group in a GDPS Continuous Availability with zero data loss environment. Records are returned until the end-of-scope log point is reached.

X'02' (WQALLCR2)
All types of log records from the proxy data sharing group in a GDPS Continuous Availability with zero data loss environment. Records are returned until the end-of-scope log point is reached.

X'03' (WQALLCR3)
Only log records for changed data capture and unit of recovery control from the proxy data sharing group in a GDPS Continuous Availability with zero data loss environment. Records are returned until the end-of-log point is reached for all members of the data sharing group.

X'04' (WQALLCR4)
All types of log records from the proxy data sharing group in a GDPS Continuous Availability with zero data loss environment. Records are returned until the end-of-log point is reached for all members of the data sharing group.

PSPI

Related tasks

Modifying Db2 for the GDPS Continuous Availability with zero data loss solution (Db2 Administration Guide)

Related reference

CDDS_MODE in macro DSN6LOGP (Db2 Installation and Migration)
CDDS_PREFIX in macro DSN6LOGP (Db2 Installation and Migration)
-START CDDS (Db2) (Db2 Commands)
-STOP CDDS (Db2) (Db2 Commands)
Qualification fields for READS requests (Db2 Performance)
Syntax and options of the REORG TABLESPACE control statement (Db2 Utilities)

Chapter 4. Ease of use

Db2 11 provides several enhancements that improve the ease of use of Db2 for z/OS, such as support for archive tables and improvements to optimization statistics feedback and statistics collection.

Improved data validation after running DSN1COPY

To prevent abends, data corruption, and storage overlays after the DSN1COPY utility runs, Db2 11 automatically validates the target data set. After a data set is populated by DSN1COPY, the first time that it is physically opened by an operation other than a utility, Db2 checks for certain data and catalog inconsistencies.

If DSN1COPY is not used correctly, subsequent attempts to access the data can result in abends, data corruption, and storage overlays. For example, the following scenarios might result in such errors:

- Incorrect DBID, PSID, or OBID values are specified when you run DSN1COPY with the OBIDXLAT option.
- DSN1COPY is used to copy data to a table space of a different type. For example, data is copied from a segmented table space to a partition-by-growth table space.
- DSN1COPY is used to copy data to a table space with a different version number or table schema.

In earlier versions of Db2, these data integrity errors might not be detected until a system failure occurs after the data is stored in the target table space. As a result, the data is unavailable.

Beginning in Db2 11, Db2 automatically detects these inconsistencies the first time that the data set is physically opened after being populated by DSN1COPY. Specifically, Db2 checks for any data and catalog inconsistencies for the following items and reports them with a -904 SQL code:

- DBID, PSID, and OBID
- SEGSIZE and PAGESIZE
- Table space type
- Table schema (Db2 checks this item if the table space contains only one table, and an OBDREC is stored in the system page.)

Several exception situations exist. Db2 does not check for data and catalog inconsistencies during the following situations:

- The data set is physically opened by a utility, including the REPAIR utility.
- Db2 is restarting.
- The header page is not formatted yet.
- The REPAIR utility is operating on the header page. (The REPAIR utility closes the page set when it is finished. Therefore, validation can be done the next time that the data set is physically opened.)
- The LOGAPPLY phase of the RECOVER utility is processing.

By not checking for inconsistencies during these situations, Db2 limits any performance impact.

If any inconsistencies are reported, you can then correct them by using the REPAIR utility with the new CATALOG option. You can also use the REPAIR utility to proactively check for any inconsistencies after you run DSN1COPY instead of waiting for the data set to be physically opened. In this case, use the REPAIR utility with the CATALOG TEST option.

Restriction: None of these data validation enhancements apply to LOB and XML table spaces.

Related reference

DSN1COPY (Db2 Utilities)

Syntax and options of the REPAIR control statement (Db2 Utilities)

© Copyright IBM Corp. 2013, 2019

Ability to alter limit keys for materialized query tables

In Db2 11, you can alter limit keys for materialized query tables or tables that are referenced by materialized query tables.

Introductory concepts

Creation of materialized query tables (Introduction to Db2 for z/OS)

In earlier versions of Db2, you cannot alter limit keys in materialized query tables or tables that are referenced by materialized query tables. You must unload the data, drop the table, re-create the table with the new partition limit key, and then reload the data.

In Db2 11, you can change the limit key for materialized query tables or tables that are referenced by materialized query tables just as you would for other tables. The only restriction is that the table must be in a partition-by-range table space or in a table space that partitioned (non-universal) with table-controlled partitioning. If this requirement is met, you can use an ALTER TABLE statement to change the limit key. In most cases, changing the limit key is a pending definition change. Changing the limit key has no impact to data availability.

Related concepts

Improved data availability when altering limit keys
In Db2 11, you can alter the limit keys for partitioned table spaces without impacting the availability of the data.

Related reference

ALTER TABLE (Db2 SQL)

Improvements to DROP PENDING CHANGES

You can remove pending table space definition changes that are not yet materialized by using the ALTER TABLESPACE statement with the DROP PENDING CHANGES clause. In Db2 11, this statement now resets advisory REORG-pending (AREOR) status.

In earlier versions of Db2, this statement does not reset the AREOR status for the affected object, even though it removes the pending changes. Therefore, if you use automated tools, this remaining AREOR status might trigger unnecessary REORG utility jobs for objects that no longer have any pending changes.

In Db2 11, an ALTER TABLESPACE statement with the DROP PENDING CHANGES clause resets any AREOR status. The only exception is hash tables that are still in a transition state to become a hash table. (The table was altered from non-hash to hash without an online REORG.) In this case, AREOR status is not reset.

Related tasks

Altering tables for hash access (deprecated) (Db2 Administration Guide)

Related reference

REORG-pending status (Db2 Utilities)
ALTER TABLESPACE (Db2 SQL)

Support for archive tables

In Db2 11, Db2 can automatically insert rows that are deleted from one table into a separate table that is called an *archive table*. Archive tables are useful for managing historical data.

You can specify that you want a table to use an archive table by specifying the new ENABLE ARCHIVE clause on the ALTER TABLE statement. The original table is called an *archive-enabled table*. You then use a global variable (SYSIBMADM.MOVE_TO_ARCHIVE) to control whether Db2 inserts rows that are deleted from this table into the associated archive table.

When you query an archive-enabled table, you can indicate whether the query considers the rows in the archive table. You also use a global variable (SYSIBMADM.GET_ARCHIVE) to indicate this preference.

Therefore, you can easily change the query to include or exclude archive table data without having to update the SQL.

Archive tables provide the following benefits:

- Db2 can manage historical data for you. You do not have to manually move data to a separate table.
- Because rows that are infrequently accessed are stored in a separate table, you can potentially improve the performance of queries against the archive-enabled table.
- You can modify queries to include or exclude archive table data without having to change the SQL statement and prepare the application again. Instead, you can control the scope of the query with a global variable.
- You can store archive tables on a lower-cost device to reduce operating costs.

Related concepts

Archive-enabled tables and archive tables (Introduction to Db2 for z/OS)

Related reference

ALTER TABLE (Db2 SQL)

Built-in global variables (Db2 SQL)

Related information

Managing Ever-Increasing Amounts of Data with IBM Db2 for z/OS: Using Temporal Data Management, Archive Transparency, and the IBM Db2 Analytics Accelerator for z/OS (IBM Redbooks)

Optimization statistics feedback

In Db2 11, Db2 identifies statistics values that are missing or conflicting when it selects access paths for SQL statements. Db2 externalizes data about the missing or conflicting statistics as optimization feedback in certain catalog and EXPLAIN tables.

Db2 externalizes data about the missing or conflicting statistics to the following tables:

- The DSN_STAT_FEEDBACK table, when you capture EXPLAIN information
- The SYSIBM.SYSSTATFEEDBACK catalog table, at the specified statistics interval

You can use the feedback data to identify missing or conflicting statistics to collect. Optimization tools can also use the optimization feedback data to recommend the collection of such statistics.

Related tasks

Identifying missing or conflicting statistics (Db2 Performance)

Related reference

SYSSTATFEEDBACK catalog table (Db2 SQL)

DSN_STAT_FEEDBACK (Db2 Performance)

Statistics collection enhancements

In Db2 11, certain utilities and commands are enhanced to improve the ease of collecting and maintaining statistics for your database objects.

Db2 11 includes the following enhancements that improve statistics collection:

- When you specify the USE PROFILE keyword in a RUNSTATS utility control statement, a default profile is applied when no profile exists for a target table.
- You can use the RUNSTATS utility to reset the access path statistics to default values for a target table space.
- You can use the ACCESS DATABASE command to immediately externalize in-memory real-time statistics to the catalog tables.
- You can collect distribution statistics when you collect inline statistics when you run the following utilities:
  - LOAD

  - REORG TABLESPACE
- You can collect histogram statistics when you collect inline statistics when you run the following utilities:
  - LOAD
  - REBUILD INDEX
  - REORG INDEX
  - REORG TABLESPACE
- You can use information that Db2 externalizes during access path selection to identify missing and conflicting statistics to collect.

Related concepts

Optimization statistics feedback

In Db2 11, Db2 identifies statistics values that are missing or conflicting when it selects access paths for SQL statements. Db2 externalizes data about the missing or conflicting statistics as optimization feedback in certain catalog and EXPLAIN tables.

Related tasks

Collecting statistics by using statistics profiles (Db2 Performance)

Updating real-time statistics immediately (Db2 Performance)

Collecting histogram statistics (Db2 Performance)

Resetting access path statistics (Db2 Utilities)

Related reference

LOAD (Db2 Utilities)

RUNSTATS (Db2 Utilities)

REBUILD INDEX (Db2 Utilities)

REORG INDEX (Db2 Utilities)

REORG TABLESPACE (Db2 Utilities)

-ACCESS DATABASE (Db2) (Db2 Commands)

Setting special registers with a profile

In Db2 11, you can now set the value of many special registers for remote applications by creating a profile in the profile tables.

You can set the value of special registers that are referenced by remote applications without changing application code. For example, you can set the level of the APPLICATION COMPATIBILITY special register to change the behavior of dynamic SQL statements without changing your application.

In applications that match the profile, special register values are set before SQL statements are run in your remote application. To set a special register, you insert into the DSN_PROFILE_ATTIBUTES table the matching PROFILEID, the KEYWORDS column value "SPECIAL_REGISTER", and the ATTRIBUTE1 column, which contains the SET special register statement.

Related tasks

Setting special registers with profile tables (Db2 Administration Guide)

Related reference

DSN_PROFILE_ATTRIBUTES profile table (Db2 Performance)

Rules for setting special registers in profiles (Db2 SQL)

Chapter 5. Security and regulatory compliance

Db2 11 continues to offer improvements to security and regulatory compliance. With key enhancements to exit authorization checking and program authorization, Db2 for z/OS and IBM Z continue to lead the industry in security and auditing.

Db2 enhancements for exit authorization checking

In Db2 11, Db2 provides the access control environment element (ACEE) of the package owner for authorization checking when the access control authorization exit is active. Db2 also refreshes the cache entries of the package authorization, the routine authorization, the DDF user authorization, and the dynamic statement when a user profile or resource access is changed in RACF and the access control authorization exit is active.

PSPI

If the AUTHEXIT_CHECK subsystem parameter is set to DB2, Db2 provides the ACEE of the package or plan owner to perform authorization checking during the processing of the autobind, BIND, REBIND PACKAGE, and REBIND PLAN commands and, if needed, the execution of the package or plan. Db2 provides the ACEE of the authorization ID as determined by the DYNAMICRULES option to perform dynamic SQL authorization checking. The access control authorization exit uses the ACEE for XAPLUCHK for authorization checking. The XAPLUCHK authorization ID can be a user or a group in RACF. To ensure successful authorization checks with the owner ACEE, the owner authorization ID in XAPLUCHK must be permitted access to the resources in RACF.

When the AUTHEXIT_CACHEREFRESH system parameter is set to ALL and the access control authorization exit is active, Db2 listens to type 62, type 71, and type 79 ENF signals from RACF for user profile or resource access changes. Then, Db2 refreshes the Db2 cache entries as needed. If you define RACF classes for Db2 objects and administrative authorities without using IBM-supplied RACF resource classes, you must enable the SIGNAL=YES option for these classes in the RACF Class Descriptor Table.

The RACF access control module (DSNXRXAC) is also enhanced to support the new global variable READ (READAUTH) and WRITE (WRITEAUTH) privileges. In addition, it returns the RACLISTED classes in the new XAPLCLST field at Db2 startup, and supports authorization checking that is associated with autobind requests for user-defined functions. PSPI

Related reference

AUTH EXIT CACHE REFR (AUTHEXIT_CACHEREFRESH subsystem parameter) (Db2 Installation and Migration)

AUTH EXIT CHECK (AUTHEXIT_CHECK subsystem parameter) (Db2 Installation and Migration)

Authorization IDs and ACEEs (Managing Security)

Db2 enhancements for program authorization

Starting in Db2 11, Db2 provides the capability to check whether an application program is authorized to use a plan.

When program authorization is enabled, it is performed in addition to package authorization.

To use package authorization effectively by itself, you need to know in advance which programs or packages use a given plan. If you do not have this information, you need to bind the plan with all collections or packages that are used by any programs that execute the plan. A disadvantage to this approach is that if a user has EXECUTE authority to run the plan, the user can accidentally invoke the wrong program, or can change the program to execute a different set of packages from the intended set.

Program authorization does not have this disadvantage, because it allows a program to run only the plans that you specify.

Program authorization is enabled for a program and its plan if the following conditions are true:

© Copyright IBM Corp. 2013, 2019

- The plan is bound with the new PROGAUTH(ENABLE) option for BIND PLAN and REBIND PLAN.
- Table SYSIBM.DSNPROGAUTH contains a row for the program and the plan.

Auditing enhancements

In DB2 10, you could use system-period temporal tables to have Db2 track when the data was modified. In Db2 11, Db2 expands its auditing capabilities to track who modified the data and the SQL statement that modified the data.

In Db2 11, tables can have *non-deterministic generated expression columns*. These columns can contain values that are helpful for auditing purposes, such as the data change operation that was performed and the value of the CURRENT SQLID and SESSION_USER special registers at the time that the data was changed.

Although this auditing enhancement is intended to complement the functionality that is already available with temporal tables, you can use this new auditing capability with any tables.

Related concepts

Scenario for tracking auditing information (Db2 Administration Guide)

Temporal tables and data versioning (Db2 Administration Guide)

Chapter 6. New application features

Db2 11 provides expanded support for SQL, XML, and temporal tables, which results in improved application performance.

Autonomous procedures

Db2 11 introduces autonomous procedures that commit independently from the calling application.

Autonomous procedures execute under their own units of work, separate from the calling program. They follow the rules of the COMMIT ON RETURN YES option when they finish, without committing the work of the calling program. The calling application program controls when its own updates are committed or rolled back.

You can define autonomous procedures by specifying the AUTONOMOUS keyword when you issue a CREATE PROCEDURE statement to create a native SQL procedure, or when you issue an ALTER PROCEDURE statement to modify an existing native SQL procedure.

Related concepts

Autonomous procedures (Db2 Application programming and SQL)

Related tasks

Controlling autonomous procedures (Db2 Administration Guide)

Related reference

ALTER PROCEDURE (SQL - native) (Db2 SQL)

CREATE PROCEDURE (Db2 SQL)

Support for explicit period specification on views

Db2 11 extends temporal support by enabling you to specify period specifications and period clauses for views. This enhancement makes implementing temporal tables easier and improves SQL consistency across the Db2 family of products.

Now, you can specify a period specification following the name of a view in the FROM clause of a query. When you query a view that references a system-period temporal table, an application-period temporal table, or a bitemporal table, you can specify a point in time or time range for a system period (SYSTEM_TIME) or an application period (BUSINESS_TIME). With this enhancement, you experience the same behavior whether you are querying a base table or a view.

Also, you can specify a period clause following the name of a target view in an UPDATE or DELETE statement. For example, you can specify a period of time for a BUSINESS_TIME period for an update or delete operation on a view that references an application-period temporal table or a bitemporal table. As a result, you experience the same behavior for the data change operation whether the operation is on a base table or a view.

Additional enhancements help to minimize the invalidation of packages for temporal tables, even if you alter a table that is referenced in a view definition and a statement that references the view is bound in a package.

Related tasks

Querying views that reference temporal tables (Db2 Administration Guide)

Changing data by using views that reference temporal tables (Db2 Administration Guide)

© Copyright IBM Corp. 2013, 2019

Improvements to querying temporal tables

In Db2 11, you can retrieve data from temporal tables for different points in time without modifying the SQL statement. Instead of changing the query, you can now use special registers to specify different points in time and run the same query.

The two new special registers are CURRENT TEMPORAL BUSINESS_TIME and CURRENT TEMPORAL SYSTEM_TIME. Both of these special registers set timestamp values to be used in queries against temporal tables. Use the CURRENT TEMPORAL BUSINESS_TIME for queries against application-period temporal tables. Use CURRENT TEMPORAL SYSTEM_TIME for queries against system-period temporal tables. When you set one of these special registers, Db2 returns rows as if the query included one of the following period specification clauses:

- FOR SYSTEM_TIME AS OF CURRENT TEMPORAL SYSTEM_TIME
- FOR BUSINESS_TIME AS OF CURRENT TEMPORAL BUSINESS TIME

In DB2 10, if you wanted to retrieve data from a temporal table for a different point time, you had to modify the query.

GUPI

For example, assume the following scenario: you have three system-period temporal tables, STT1, STT2 and STT3, and you have an application that contains the following SQL statement:

```
SELECT UDF_1(STT1.C1) FROM STT1,STT2
```

The user-defined function UDF_1 consists of the following SQL statement:

```
SELECT * FROM STT3
```

Today is 2012-03-26-17.44.49.000000, and the application wants to find the result from last week.

In DB2 10, you had to take the following steps:

1. Modify the SQL statement as follows:

   ```
   SELECT UDF_(STT1.C1) FROM
   STT1 FOR SYSTEM_TIME AS OF  '2012-03-19-17.44.49',
   STT2 FOR SYSTEM_TIME AS OF  '2012-03-19-17.44.49'
   ```

2. Modify the SQL in UDF_1 as follows:

   ```
   SELECT * FROM
   STT3 FOR SYSTEM_TIME AS OF  '2012-03-19-17.44.49'
   ```

3. Precompile, compile, and bind both the application and UDF_1.

In Db2 11, set the CURRENT TEMPORAL SYSTEM_TIME special register before you call the application:

```
SET CURRENT TEMPORAL SYSTEM_TIME = '2012-03-19-17.44.49'
```

You do not have to change the application or UDF_1. Alternatively, if you want the active data, set the special register to NULL, which is the default value for the special register.

GUPI

Related tasks

Querying temporal tables (Db2 Administration Guide)

Related reference

CURRENT TEMPORAL BUSINESS_TIME (Db2 SQL)

CURRENT TEMPORAL SYSTEM_TIME (Db2 SQL)

Related information

Managing Ever-Increasing Amounts of Data with IBM Db2 for z/OS: Using Temporal Data Management, Archive Transparency, and the IBM Db2 Analytics Accelerator for z/OS (IBM Redbooks)

XML enhancements

Db2 11 provides enhanced support for XML data.

Db2 implicitly adds a document node when it stores data with the following statements:

- INSERT
- UPDATE
- XMLDOCUMENT

Db2 issues fewer error SQLCODEs when it evaluates XPath predicate expressions with an explicit cast or an operation with an invalid value. If the XML data that is processed is filtered from the result, then processing the query continues. Examples of XPath expressions that have fewer errors include the following situations:

- When data is filtered from the result by the predicate before an invalid operation. For example, an operation calculates a division of a number by zero.
- When data is explicitly cast to an incompatible data type.

The LOAD utility performance is improved when it loads binary XML data that has been previously validated. When certain conditions are met, you might see improvements to CPU time when you are loading binary XML data that was created with the UNLOAD utility.

Related concepts

Best practices for XML performance in Db2 (Db2 Performance)

XML support for the cross-loader function

Db2 11 provides support for the XML data type for the cross-loader function.

This enhancement improves data portability by enabling you to use the LOAD utility with the INCURSOR option to move XML data from a local or remote table to a local table. In addition, for the cross-loader function, the limitation on the sum of LOB column lengths is removed.

Related tasks

Loading data by using the cross-loader function (Db2 Utilities)

Support for Java stored procedures in a 64-bit JVM

Db2 11 introduces support for running Java™ stored procedures in a 64-bit JVM.

Earlier versions of Db2 can run Java stored procedures in 31-bit Java virtual machines (JVMs) only, and each JVM can run only one Java stored procedure at a time. Db2 11 can concurrently run multiple Java stored procedures in 64-bit JVMs. Therefore, in Db2 11, more Java stored procedures can run in a single stored procedure address space than in earlier versions.

Tip: For WLM environments that run 64-bit JVMs, set the NUMTCB parameter to 25. For WLM environments that run 31-bit JVMs, set the NUMTCB parameter to 5.

Related tasks

Setting up the environment for Java routines (Db2 Application Programming for Java)

Support for variable-length timestamps in ODBC applications

In Db2 11, ODBC applications can retrieve and update data TIMESTAMP data with up to 12 fractional digits of precision.

In earlier versions of Db2, ODBC applications can update and retrieve timestamp data with up to microseconds of precision (six decimal digits). In Db2 11, ODBC applications can update and retrieve timestamp data with up to picoseconds of precision (12 decimal digits).

Related concepts

Variable-length timestamps in ODBC applications (Db2 Programming for ODBC)

Support for arrays

Db2 11 includes array support in SQL statements.

This support makes it easier to exchange long lists of values with the data server. Array support on Db2 for z/OS is compatible with the support that is offered on other members of the Db2 family of products.

With array support in Db2 11 you can do the following tasks:

- Define arrays as parameters and variables for SQL routines
- Pass arrays from one procedure to another as arguments for input or output parameters
- Pass arrays to functions as parameters or from functions as return values
- Manipulate arrays, transform arrays to tables, and transform tables to arrays by using new built-in functions

Related concepts

Array types (Db2 SQL)

Expanded support for not-logged table spaces

In Db2 11, you can specify the logging attributes LOGGED or NOT LOGGED when you define declared global temporary tables. The NOT LOGGED option suspends logging during insert, update, and delete activity for the table.

Because the existing logging facilities in Db2 are finely tuned, suspending logging does not necessarily improve the performance of your system. However, the ability to suspend logging improves scalability, particularly for operations that insert large volumes of data.

Related reference

DECLARE GLOBAL TEMPORARY TABLE (Db2 SQL)

Enhancement to the LIKE predicate

In Db2 11, enabling the new LIKE_BLANK_INSIGNIFICANT subsystem parameter enhances the LIKE predicate so that blanks at the end of fixed-length strings are ignored. Because of this new behavior, results for the LIKE predicate are more consistent regardless of whether the column data contains fixed-length strings or variable-length strings.

This behavior is called *LIKE blank insignificant* behavior. *LIKE blank significant* behavior, in which the blanks at the end of fixed-length strings are significant (not ignored), is the default behavior during installation or migration. For variable-length strings, blanks are significant.

Tip: After you enable the LIKE_BLANK_INSIGNIFICANT subsystem parameter, existing rows might not conform to table check constraints that contain a LIKE predicate. Consider running the CHECK DATA utility on all affected tables to find the records that do not conform to the table check constraint.

Related reference

LIKE predicate (Db2 SQL)

LIKE BLANK INSIGNIFICANT field (LIKE_BLANK_INSIGNIFICANT subsystem parameter) (Db2 Installation and Migration)

Support for user-defined global variables

Db2 11 introduces support for user-defined global variables, which you can use to share relational data between SQL statements without the need for additional application logic.

A *user-defined global variable* is a global variable that is defined by the user at the database manager level. User-defined global variables help to maximize the flexibility of a database management system by providing the following benefits:

- Faster and easier porting of applications from the databases of vendor software

- Reduced cost of ownership for Db2 for z/OS
- Easier implementation of Db2 for z/OS applications

Related reference
CREATE VARIABLE (Db2 SQL)

Column masking improvements

In Db2 11, column access control restrictions for the GROUP BY clause and the HAVING clause are removed. This improvement is beneficial when you implement row and column access control.

In certain contexts in DB2 10, the semantics of the column mask can conflict with the semantics in the GROUP BY clause or the HAVING clause. When these conflicts occur, the column mask cannot be applied for the statement and an error is returned at bind time. However, now when the GROUP BY or HAVING clauses are used, columns with column masks can be referenced in the result table without any restrictions.

Also, inconsistencies between the behavior of the SELECT DISTINCT statement and an aggregate function with the DISTINCT keyword are resolved. Now, column masking for an aggregate function with the DISTINCT keyword is applied after the removal of duplicate rows.

Related tasks
Coding SQL statements to avoid unnecessary processing (Db2 Performance)
Excluding duplicate rows from the result table of a query (Db2 Application programming and SQL)
Related reference
select-clause (Db2 SQL)

Transform XML with an XSL style sheet

The user-defined function, XSLTRANSFORM, provides support for transforming an XML document with XSL Transformation processing.

The Java user-defined function, SYSFUN.XSLTRANSFORM, transforms well-formed XML documents that are stored in Db2 using the XL TXE-J processor in the IBM SDK for z/OS. Input parameters are as follows:

1. An expression that returns a single-rooted, well-formed XML document.
2. An XSL style sheet that conforms to the W3C XSLT Version 1.0 Recommendation.
3. An expression that returns an XML document that contains parameter values to the XSL style sheet.

The output of the user-defined function is a CLOB.

Related concepts
Transforming an XML document with XSLTRANSFORM (Db2 Programming for XML)
Related reference
XSLTRANSFORM (Db2 SQL)

RRSAF connection functions that support longer client information fields

Some RRSAF connection functions now support longer names for user, application name, workstation name, and correlation token.

You can now specify user up to 128 bytes, application name up to 255 bytes, workstation name up to 255 bytes, and correlation token up to 255 bytes. The following RRSAF connection functions now have parameters that support longer client information fields:

- SIGNON function for RRSAF
- AUTH SIGNON function for RRSAF
- SET_CLIENT_ID function for RRSAF

Related reference

SIGNON function for RRSAF (Db2 Application programming and SQL)
AUTH SIGNON function for RRSAF (Db2 Application programming and SQL)
SET_CLIENT_ID function for RRSAF (Db2 Application programming and SQL)

Support for temporal auditing

Db2 11 adds support for temporal auditing to enhance the automatic management of data versioning and tracking.

The temporal auditing enhancements introduced in Db2 11 add integrated auditing support to allow for the automatic tracing of the following types of audit information: who modified the data, when the data were modified, and what SQL operation modified the data in the table.

Related concepts

Creation of temporal tables (Introduction to Db2 for z/OS)

DevOps with Db2: Automated deployment of applications with IBM UrbanCode Deploy

With UrbanCode® Deploy, you can easily automate the deployment and configuration of database schema changes in Db2 11 and Db2 12. The automation reduces the time, costs, and complexity of deploying and configuring your business-critical apps, getting you to business value faster and more efficiently.

Db2 for z/OS - IBM Urban Code Deploy - Automate Application Deployments
IBM Db2 for z/OS Package (UrbanCode Deploy)
UrbanCode Deploy Tour
Deploy database applications against Db2 for z/OS with IBM UrbanCode deploy

Related concepts

Db2 application provisioning, development, and deployment (DevOps) solutions ()

Related information

UrbanCode Deploy

Modern language support Db2 for z/OS application development

Db2 11 and Db2 12 now support application development in many modern programming and scripting languages. Application developers can use languages like Python, Perl, and Ruby on Rails to write Db2 for z/OS applications. Getting business value from your mainframe applications is now more accessible than ever before.

Take advantage of Db2 for z/OS modern application development language support
IBM data server client and driver types
IBM Data Server Driver Package (Windows)
Programming for Db2 for z/OS (Introduction to Db2 for z/OS)

Db2 REST services improve efficiency and security

The Db2 REST services support, available in Db2 11, unleashes your enterprise data and applications on Db2 for z/OS for the API economy. Mobile and cloud application developers can use efficiently created, consumable, scalable, and RESTful services by using any REST client. They can use these services to securely interact with business-critical data and transactions, without special Db2 for z/OS expertise.

Recent service updates to the Db2 REST services support introduces the BIND SERVICE and FREE SERVICE commands and versioning to REST services. Db2 database administrators can issue these DSN commands to create and drop user-defined REST services the same way as they use BIND PACKAGE and FREE PACKAGE for normal application package creation and deletion. The Db2 commands, START RESTSVC, STOP RESTSVC, and DISPLAY RESTSVC, were also introduced with the recent service updates

to the Db2 REST services support. The commands can be used by DBAs to manage the availability of user-defined REST services.

Db2 for z/OS RESTful API enabling the Mobile Economy
Db2 for z/OS Native REST services (IBM developerWorks)
Accessing HTTP and RESTful services from Db2: Introducing the REST user-defined functions for Db2 (IBM developerWorks)

Related concepts
Db2 REST services ()
Related reference
FREE SERVICE (DSN) (Db2 Commands)
BIND and REBIND options for packages, plans, and services (Db2 Commands)

Chapter 7. Migration

As with each version of Db2 for z/OS, Db2 11 continues to provide substantial improvements that make migration easier and faster. Db2 11 provides enhancements to the Db2 installation CLIST, ISPF panels, and jobs, and provides new installation verification procedures (IVPs).

Improved reliability and contention relief for online migration

In Db2 11, you can identify applications that might be incompatible with catalog migrations and updates.

Before scheduling an online migration, you can issue the new DISPLAY BLOCKERS command or invoke the new BLOCKING_THREADS built-in function to identify locks and claims that might interfere with migration processing. This information allows you to take appropriate actions beforehand to minimize the possibility of a failed migration.

The new command and function can be used for both system and non-system databases. For more information, see Identify applications that are incompatible with online migration (Db2 Installation and Migration).

Related reference

-DISPLAY BLOCKERS (Db2) (Db2 Commands)

BLOCKING_THREADS (Db2 SQL)

Improvements to Db2 installation and samples

Db2 11 continues to provide improvements to installing and migrating Db2 subsystems and data sharing groups. This version of Db2 for z/OS provides enhancements to the Db2 installation CLIST, ISPF panels, and jobs, and provides new installation verification procedures (IVPs).

The Db2 installation and migration jobs have the following enhancements:

- You can now specify functional IDs for running the installation, migration, and verification jobs. In installation panel DSNTIPG, you can specify the following functional IDs for use in the installation and IVP jobs:
  - A functional ID that is used as the current SQLID
  - A functional ID that is used as the OWNER value for the BIND commands
  - A functional ID to which privileges are granted on objects that are used by the installation jobs
- In installation panel DSNTIPG, you can specify an ID under which Db2-supplied routines are created and bound. You can also specify an ID that is used as the SECURITY DEFINER value for Db2-supplied routines.
- The installation CLIST now supports a maximum length of 44 bytes for Db2 target library names.
- BIND PLAN commands in the installation, migration, and IVP jobs now contain ACTION(REPLACE) RETAIN, so that EXECUTE privileges are preserved when the plans are bound again.

Related concepts

The Db2 installation CLIST panel session (Db2 Installation and Migration)

Related tasks

Tailoring Db2 11 installation and migration jobs with the CLIST (Db2 Installation and Migration)

Related reference

DSNTIPG: Installation preferences panel (Db2 Installation and Migration)

© Copyright IBM Corp. 2013, 2019

Application compatibility by version

Db2 11 adds the ability to set the SQL function and features available to an application package based on an application compatibility value.

In Db2 11 new-function mode, you can run your application with either V10R1 or V11R1 features and functions. If you migrate your Db2 environment to Db2 11 new-function mode, you can continue to run individual applications with DB2 10 SQL features and functions. You can control which of your applications to run with V10R1 and which to run with V11R1. Leaving applications set to V10R1 gives you time to review differences in behavior between versions. The value of application compatibility is based on a bind option of your package, a special register, or a subsystem parameter value.

Related concepts

Application compatibility levels in Db2 (Db2 Application programming and SQL)

Part 2. What's changed in Db2 11

Use this information when you are planning migration to Db2 11 and for planning to adopt new capabilities that Db2 11 introduces.

Db2 for z/OS News from the Lab blog: See the Db2 for z/OS News from the Lab blog for the latest news about new capabilities and enhancements in Db2 for z/OS continuous delivery, from the IBM experts who design, build, test, and support Db2

Subscribe today!

This section contains information about incompatible changes that might impact your migration to Db2 11. It also contains summaries of the changes that Db2 11 introduces. It also summarizes function the Db2 11 no longer supports, and deprecated function that remains supported, but might be removed eventually.

Related reference

New, changed, and deleted codes (Db2 Codes)

New, changed, and deleted messages (Db2 Messages)

© Copyright IBM Corp. 2013, 2019

Chapter 8. Incompatible changes for migration to Db2 11

Before migrating your Db2 10 subsystem to Db2 11, familiarize yourself with incompatible changes that might impact your migration. Plan to resolve any such applicable incompatible changes that apply to your Db2 environment before or during the Db2 11 migration process.

Related concepts

Function that Db2 11 no longer supports
If you are migrating to Db2 11 from Db2 10, be aware of function that is no longer supported.

What's changed in Db2 11
Use this information when you are planning migration to Db2 11 and for planning to adopt new capabilities that Db2 11 introduces.

Related reference

Deprecated function in Db2 11
Certain capabilities that Db2 11 for z/OS supports are *deprecated*, meaning that their use is discouraged. Although they remain supported except as noted below in Db2 11, support is likely to be removed eventually.

Application and SQL release incompatibilities

When you migrate from Db2 10 to Db2 11, be aware of and plan for application and SQL release incompatibilities that might affect your migration.

re

GUPI

Plan for the following changes in Db2 11 that might affect your migration.

Release incompatibilities that were changed or added since the first edition of this Db2 11 publication are indicated by a vertical bar in the left margin. In other areas of this publication, a vertical bar in the margin indicates a change or addition that has occurred since the Db2 10 release of this publication.

Change to determination of ASUTIME for dynamic statements

In Db2 11 new-function mode with application compatibility set to 'V11R1', the dynamic SQL ASUTIME limit for each routine is used by the resource limit facility.

Explanation

The ASUTIME limit that is specified for the routine determines the limit. If the dynamic SQL statements in a routine use more ASUTIME than the limit, then SQLCODE -905 is returned. This SQLCODE occurs even if the value is lower than the ASUTIME limit of a top-level calling package. The ASUTIME limit that is specified for the top-level calling package is not considered. In previous versions of Db2, SQLCODE -905 is issued when the limit of the top-level calling package is encountered. Now in Db2 11 the time that is used by each nested routine is compared to the ASUTIME limit.

Possible impact to your Db2 environment

Because the limit is enforced for each monitored routine, your applications might return more SQLCODE -905 errors.

Actions to take

While in conversion mode with application compatibility for your package set to value 'V10R1', run your applications with IFCID 0366 or IFCID 0376 enabled. Then, review the trace output for incompatible

© Copyright IBM Corp. 2013, 2019

changes with the identifier ’1103’. Review and, if necessary, adjust the ASUTIME limits on routines and packages that use dynamic SQL.

Related concepts

Application compatibility levels in Db2 (Db2 Application programming and SQL)

Related tasks

Setting limits for system resource usage by using the resource limit facility (Db2 Performance)

Related information

-905 (Db2 Codes)

Automatic rebind of plans and packages created before DB2 9

Explanation

Plans and packages that were last bound before DB2 9 are not supported in Db2 11 conversion mode and later.

Possible impact to your Db2 environment

If you specify YES or COEXIST for the ABIND subsystem parameter, Db2 11 automatically rebinds plans and packages that were bound before DB2 9. As a result, an execution delay might occur the first time that such a plan or package is loaded. Also, the automatic rebind might change the access path to a potentially more efficient access path.

If you specify NO for the ABIND subsystem parameter, negative SQLCODEs are returned for each attempt to run a package or plan that was bound before DB2 9. SQLCODE -908, SQLSTATE 23510 is returned for packages, and SQLCODE -923, SQLSTATE 57015 is returned for plans until they are rebound in Db2 11.

Actions to take

To identify plans and packages that were bound before DB2 9, run the Db2 11 premigration job DSNTIJPM on your DB2 10 catalog.

Invalidated plans and packages

During the enabling-new-function mode processing, plans and packages that reference the affected Db2 catalog and directory table spaces become invalidated.

Explanation

The following table spaces in the Db2 catalog and directory are modified when you run job DSNTIJEN to enable Db2 11 new-function mode:

- DSNDB01.SYSUTILX
- DSNDB01.SYSLGRNX
- DSNDB06.SYSCOPY
- DSNDB06.SYSRTSTS
- DSNDB06.SYSTSIXS
- DSNDB06.SYSTSTAB
- DSNDB06.SYSSTR

As these table spaces are processed, Db2 invalidates packages or plans that reference them.

The packages that are dependent on the following catalog tables are also invalidated:

- SYSIBM.SYSCOPY
- SYSIBM.SYSCHECKS
- SYSIBM.SYSCHECKS2
- SYSIBM.SYSCHECKDEP

- SYSIBM.SYSCOLUMNS
- SYSIBM.SYSINDEXES
- SYSIBM.SYSPENDINGDDL
- SYSIBM.SYSSTRINGS
- SYSIBM.SYSTABLES
- SYSIBM.SYSTABLESPACE
- SYSIBM.SYSINDEXSPACESTATS
- SYSIBM.SYSTABLESPACESTATS

Possible impact to your Db2 environment

If you have autobind enabled, the invalid packages are bound on the first run after they were marked invalid.

If you have autobind disabled, each attempt to use an invalidated package fails with SQLCODE -908 to indicate that the application must be bound before it can be run.

Actions to take

For SYSLGRNX, existing CHAR(6) columns were changed to CHAR(10). You might need to modify your application before it can run successfully.

For SYSUTILX, the RBA fields were moved to new fields. Applications might need to be updated before you can see the new fields.

The SYSCOPY table space was replaced by a new table space, SYSTSCPY. You might need to modify your application before it can run successfully.

The SYSRTSTS table space was replaced by two new table spaces, SYSTSTSS and SYSTSISS. SYSTSTSS contains the SYSIBM.SYSTABLESPACESTATS catalog table and SYSTSISS contains the SYSIBM.SYSINDEXSPACESTATS table. You might need to modify your application before it can run successfully.

The SYSSTR table space was replaced by four new table spaces, SYSTSCKS, SYSTSCHX, SYSTSCKD, and SYSTSSRG. SYSTSCKS contains SYSIBM.SYSCHECKS, SYSTSCHX contains SYSIBM.SYSCHECKS2, SYSTSCKD contains SYSIBM.SYSCHECKDEP, and SYSTSSRG contains SYSIBM.SYSSTRINGS catalog table. You might need to modify your application before it can run successfully.

Default for ODBC limited block fetch

The default for the LIMITEDBLOCKFETCH initialization keyword changed.

Explanation

In DB2 10, ODBC limited block fetch was disabled by default. In Db2 11, ODBC limited block fetch is enabled by default.

Possible impact to your Db2 environment

Your applications might use limited block fetch, when they did not do so previously.

Actions to take

If the default is not appropriate for your ODBC applications, you can change it by modifying the value of the LIMITEDBLOCKFETCH initialization keyword.

Related reference

Db2 ODBC initialization keywords (Db2 Programming for ODBC)

Views, materialized query tables, and SQL table functions with period specifications

Explanation

In Db2 11, views, materialized query tables, and SQL table functions that were created with period specifications in DB2 10 are not supported.

Possible impact to your Db2 environment

If such views, materialized query tables, or SQL functions are used in Db2 11, incorrect results might occur.

Actions to take

To prepare for this change, drop all views, materialized query tables, and SQL table functions that contain a SYSTEM_TIME or BUSINESS_TIME period specification.

To identify such existing views, materialized query tables, and SQL table functions, run the Db2 11 premigration job DSNTIJPM on your DB2 10 catalog.

You can also manually issue the following queries.

To identify views and materialized query tables that were created with a period specification, issue the following query:

```
SELECT * FROM SYSIBM.SYSVIEWDEP WHERE BTYPE IN ('W', 'Z') AND DTYPE IN ('V', 'M');
```

To identify SQL table functions that were created with a period specification, issue the following query:

```
SELECT * FROM SYSIBM.SYSDEPENDENCIES WHERE BTYPE = 'Z';
```

To identify SQL scalar functions that were created with a period specification or period clause, issue the following query:

```
SELECT * FROM SYSIBM.SYSPACKDEP WHERE BTYPE IN ('W', 'Z') AND DTYPE = 'N';
```

Dropping columns named CLONE, ORGANIZATION, or VERSIONING

In Db2 11 new-function mode, a column that is named CLONE, ORGANIZATION, or VERSIONING should be specified as a delimited identifier in order to be dropped from a table.

Explanation

Prior to Db2 11, CLONE, ORGANIZATION, and VERSIONING are reserved keywords that can appear after the DROP keyword in an ALTER TABLE statement. When CLONE, ORGANIZATION, or VERSIONING is specified as a simple token (that is, not as a delimited identifier), these keywords can only match the DROP CLONE, DROP ORGANIZATION, or DROP VERSIONING clauses on an ALTER TABLE statement.

Possible impact to your Db2 environment

If you intend to drop a column named CLONE, ORGANIZATION, or VERSIONING in Db2 11, and the name is specified as a simple token on the ALTER TABLE statement, the Db2 subsystem might interpret the ALTER TABLE statement as specifying the DROP CLONE, DROP ORGANIZATION, or DROP VERSIONING clauses instead of the DROP COLUMN clause.

Actions to take

To drop a column named CLONE, ORGANIZATION, or VERSIONING in Db2 11, the name must be specified as a delimited identifier. For example: DROP "ORGANIZATION" or DROP "CLONE" (assuming " is the delimiter for a delimited identifier).

Alternatively, you can specify the optional COLUMN keyword in the DROP COLUMN clause. For example: DROP COLUMN ORGANIZATION or DROP COLUMN CLONE.

Related reference
ALTER TABLE (Db2 SQL)

Allow XPath processing to continue even if error on filtered results

In Db2 11 new-function mode with application compatibility set to 'V11R1', XPath processing might return fewer errors on predicate expressions with an explicit cast or an operation with an invalid value.

Explanation
In previous versions, even though the invalid result is filtered from the result set, XPath processing would return an error SQLCODE. In Db2 11, examples of XPath expressions that have fewer errors include situations when:

- Data is filtered from the result by the predicate before an invalid operation such as division of a number by zero
- Data is explicitly cast to an incompatible data type

Possible impact to your Db2 environment

Your applications might return fewer error SQLCODEs.

Actions to take

While in conversion mode with application compatibility for your package set to value 'V10R1', run your applications with IFCID 0366 or IFCID 0376 enabled. Then, review the trace output for incompatible changes with the identifier '1102'.

Related concepts
XML enhancements
Db2 11 provides enhanced support for XML data.

Application compatibility levels in Db2 (Db2 Application programming and SQL)

XML document node implicitly added on insert and update

In Db2 11 new-function mode with application compatibility set to 'V11R1', if an XML document does not have a document node, then one is added during insert and update operations.

Explanation
In previous versions of Db2, document nodes are not implicitly added and an SQL insert or update of an XML document returned SQLCODE -20345. To avoid the error, an application invokes the XMLDOCUMENT function before the insert or update. In Db2 11, an XML document node is added if one does not exist in the XML document.

Possible impact to your Db2 environment

Your applications might return fewer errors on insert and update operations.

Actions to take

While in conversion mode with application compatibility for your package set to value 'V10R1', run your applications with IFCID 0366 or IFCID 0376 enabled. Then, review the trace output for incompatible changes with the identifier '1101'. In addition, you can review your applications for use of the XMLDOCUMENT function.

Related concepts
XML enhancements
Db2 11 provides enhanced support for XML data.

Application compatibility levels in Db2 (Db2 Application programming and SQL)

Client information special registers length

In Db2 11 new-function mode with application compatibility set to 'V11R1', special registers for client information fields might return different length values. The values in special registers CURRENT CLIENT_USERID, CURRENT CLIENT_WRKSTNAME, CURRENT CLIENT_APPLNAME, and CURRENT CLIENT_ACCTNG are determined by the application compatibility level.

Explanation

In previous versions of Db2, client information values were truncated and padded to the maximum length. In Db2 11 the following lengths increase:

- The maximum length of CURRENT CLIENT_USERID increases from 16 bytes to 128 bytes.
- The maximum length of CURRENT CLIENT_WKSTNAME increases from 18 bytes to 255 bytes.
- The maximum length of CURRENT CLIENT_APPLNAME increases from 32 bytes to 255 bytes.
- The maximum length of CURRENT CLIENT_ACCTNG increases from 200 bytes to 255 bytes.

In Db2 11, trailing blanks are removed.

Possible impact to your Db2 environment

When the application compatibility for your package is set to value 'V11R1', your applications might receive a different length client information value than they did previously. The value is no longer padded to the supported maximum length and trailing blanks are removed.

Actions to take

Review your applications for use of these special registers. While in conversion mode with application compatibility for your package set to value 'V10R1', run your applications with IFCID 0366 or IFCID 0376 enabled. Then, review the trace output for incompatible changes with the identifier '1104', '1105', '1106', or '1107'.

Related concepts

Application compatibility levels in Db2 (Db2 Application programming and SQL)

Client information results from ADMIN_COMMAND_DB2

In Db2 11 conversion mode, the ADMIN_COMMAND_DB2 result set row returned changes in the created global temporary table SYSIBM.DB2_THREAD_STATUS when processing-type = "THD". The column data type and maximum lengths for WORKSTATION, USERID, APPLICATION, and ACCOUNTING change.

Explanation

In Db2 11 the following column data types and lengths change:

- WORKSTATION increases from CHAR(18) to VARCHAR(255).
- USERID increases from CHAR(16) to VARCHAR(128).
- APPLICATION increases from CHAR(32) to VARCHAR(255).
- ACCOUNTING increases from CHAR(247) to VARCHAR(255).

Possible impact to your Db2 environment

Your applications now receive a VARCHAR data type and possibly a different length client information value. The length is no longer padded to the supported maximum length.

Actions to take

Review your applications for use of the ADMIN_COMMAND_DB2 stored procedure.

Related reference

ADMIN_COMMAND_DB2 stored procedure (Db2 SQL)

ALTER statements that change limit keys are pending changes

Starting in Db2 11 new function mode, ALTER statements that modify limit key values result in pending data definition changes, which do not take effect until materialized by the REORG utility. They also block all subsequent immediate changes until materialized.

Explanation

In Db2 11, ALTER statements that change limit keys for the following types of table spaces result in pending data definition changes:

- partition-by-range spaces
- Partitioned (non-UTS) tables spaces with table-controlled partitioning

Affected partitions are placed in advisory REORG-pending (AREOR) status. In DB2 10, such ALTER statements resulted in immediate changes, which placed affected partitions in restrictive REORG-pending (REORP) status.

ALTER statements that alter the last partition are an exception if they change MAXVALUE to a value less than MAXVALUE for ascending, or from MINVALUE to a value greater than MINVALUE for descending. In such cases, the changes are immediate and affected partitions are placed in restricted REORG-pending status.

Possible impact to your Db2 environment

Affected partitions remain in advisory REORG-pending (AREOR) status, the old limit key values remain in effect, and the data remains available until the pending definition change is materialized. However, all subsequent immediate data definition changes, including in the same statement, remain blocked until the pending data definition changes are materialized. Table spaces with pending definition changes have an entry in SYSIBM.SYSPENDINGDDL.

Also, you can no longer materialize changes for ALTER statements that change limit keys by using the REORG TABLESPACE utility with SHRLEVEL NONE or the LOAD utility with REPLACE.

Actions to take

Modify any existing jobs that materialize limit key changes to run the REORG TABLESPACE utility with SHRLEVEL CHANGE or SHRLEVEL REFERENCE.

For jobs that use the LOAD utility with REPLACE, modify the job to run the REORG TABLESPACE utility with SHRLEVEL CHANGE or SHRLEVEL REFERENCE before running LOAD.

Related concepts

Improved data availability when altering limit keys
In Db2 11, you can alter the limit keys for partitioned table spaces without impacting the availability of the data.

Pending data definition changes (Db2 Administration Guide)

Related tasks

Materializing pending definition changes (Db2 Administration Guide)

Altering table spaces (Db2 Administration Guide)

Related reference

REORG-pending status (Db2 Utilities)

SYSTABLEPART.LIMITKEY format variations

Explanation

Starting in Db2 11 conversion mode, the LIMITKEY column of the SYSTABLEPART catalog table can contain a mix of differently formatted values:

- Date and time values are delimited by single quotation marks (for example, `'2001-01-01'`). However, values that were added in releases prior to Db2 11 do not contain the delimiters.
- When the decimal point indicator is comma, a space follows any comma delimiter in the value. The comma decimal point indicator is used when the DECIMAL POINT IS field setting is `, (comma)` or a COBOL program that executes the ALTER statement uses the COMMA processing option. No space follows comma delimiters when period decimal point indicators are used, or for values added in releases prior to Db2 11.

Possible impact to your Db2 environment

Applications that do not tolerate the format variations might fail.

Actions to take

Modify any applications that use the LIMITKEY column to tolerate the format variations.

For example, to remove the single-quote delimiters for date and time values, from a single column in a partitioning key, you might use the following SQL statements:

```
SELECT DATE(STRIP(LIMITKEY,B,X'27'))  FROM SYSIBM.SYSTABLEPART WHERE …;
SELECT TIME(STRIP(LIMITKEY,B,X'27'))  FROM SYSIBM.SYSTABLEPART WHERE …;
```

If `(STRIP(LIMITKEY,B,X'27'))` is omitted from the statements, Db2 issues SQLCODE -180.

The use type of the column is also changed to S, which indicates a product-sensitive programming interface. It is expected that programs written to such interfaces might need to be changed in order to run with new product releases or versions, or as a result of service.

Related reference

SYSTABLEPART catalog table (Db2 SQL)
Programming interface information (Introduction to Db2 for z/OS)
DECIMAL POINT IS field (DECIMAL DECP value) (Db2 Installation and Migration)
Descriptions of SQL processing options (Db2 Application programming and SQL)

Removing the SYSPUBLIC schema from the SQL PATH routine option

Starting in Db2 11 conversion mode, SYSPUBLIC is the schema that is used for public aliases. As such, the SQL PATH routine option must not specify the SYSPUBLIC schema.

Explanation

In previous versions of Db2, you could not define functions, procedures, distinct types, and sequences in the SYSPUBLIC schema, but you were not restricted from specifying SYSPUBLIC as part of the SQL PATH. However, doing so no effect on applications. In Db2 11 you can no longer specify SYSPUBLIC as part of the SQL PATH.

Possible impact to your Db2 environment

Creation or resolution of some objects that worked in previous versions, might fail in Db2 11 with SQLCODE -713 if SYSPUBLIC is specified as part of the SQL PATH.

Actions to take

Query the catalog to see if any object schemas use SYSPUBLIC as the schema qualifier. This is highly unlikely for any object, but most likely with objects that use the SQL PATH (functions, procedures, distinct types, and sequences).

Change any existing SET PATH statements to not specify SYSPUBLIC as a schema.

Related concepts

SQL path (Db2 SQL)
Unqualified type, function, procedure, global variable, and specific names (Db2 SQL)

Related reference
CURRENT PATH (Db2 SQL)
SET PATH (Db2 SQL)
Related information
-713 (Db2 Codes)

SYSIBMADM schema added to the SQL path

In Db2 11 new-function mode with application compatibility set to 'V11R1', SYSIBMADM is added to the SQL path as an implicit schema.

Explanation
If SYSIBMADM is not specified as an explicit schema in the SQL path, it is included as an implicit schema at the beginning of the path after SYSIBM, SYSFUN, and SYSPROC.

Possible impact to your Db2 environment

Applications that reference the content of the CURRENT PATH special register now have the SYSIBMADM schema returned when implicit schemas are included in the path. For example, the statement SELECT CURRENT PATH FROM SYSIBM.SYSDUMMY1 now returns "SYSIBM","SYSFUN","SYSPROC","SYSIBMADM","*authid*," where *authid* is the authorization ID of the statement, instead of "SYSIBM","SYSFUN","SYSPROC","*authid*."

Actions to take

No action is required.

Related concepts
SQL path (Db2 SQL)
Related reference
CURRENT PATH (Db2 SQL)

Change in result for CAST(*string* AS TIMESTAMP)

In Db2 11 new-function mode with application compatibility set to 'V11R1', the result of CAST(*string* AS TIMESTAMP) is changed in some cases.

Explanation
Previously, when Db2 executed CAST(*string* AS TIMESTAMP), Db2 interpreted an 8-byte string as a Store Clock value and a 13-byte string as a GENERATE_UNIQUE value. This interpretation might result in an incorrect result from the CAST specification. Starting with Db2 11, with the application compatibility set to V11R1, when an 8-byte string or a 13-byte string is input to CAST(*string* AS TIMESTAMP), Db2 interprets the input strings as string representations of TIMESTAMP values.

Possible impact to your Db2 environment

An invalid representation of an 8-byte or 13-byte string in CAST(*string* AS TIMESTAMP) results in SQLCODE -180.

For example, suppose that you execute the following SQL statements in Db2 11 new-function mode:

```
-- SET APPLICATION COMPATIBILITY TO V10R1
SET CURRENT APPLICATION COMPATIBILITY='V10R1';
-- CAST AN 8-BYTE STRING REPRESENTATION OF A DATETIME VALUE
-- TO TIMESTAMP
SELECT CAST('1/1/2013' AS TIMESTAMP) FROM SYSIBM.SYSDUMMY1;
-- CAST AN 8-BYTE STRING REPRESENTATION OF A STORE CLOCK VALUE
-- TO TIMESTAMP
SELECT CAST(X'CAB5060708090100' AS TIMESTAMP) FROM SYSIBM.SYSDUMMY1;
```

The result is of the first SELECT statement is 2034-07-25-16.43.41.599503, which is an incorrect result. The result of the second SELECT statement is 2013-01-01-20.37.04.246928, which is the correct result if the input string is interpreted as a Store Clock value.

If you execute the following SQL statements, the results differ:

```
-- SET APPLICATION COMPATIBILITY TO V11R1
SET CURRENT APPLICATION COMPATIBILITY='V11R1';
-- CAST AN 8-BYTE STRING REPRESENTATION OF A DATETIME VALUE
-- TO TIMESTAMP
SELECT CAST('1/1/2013' AS TIMESTAMP) FROM SYSIBM.SYSDUMMY1;
-- CAST AN 8-BYTE STRING REPRESENTATION OF A STORE CLOCK VALUE
-- TO TIMESTAMP
SELECT CAST(X'CAB5060708090100' AS TIMESTAMP) FROM SYSIBM.SYSDUMMY1;
```

The result of the first SELECT statement is 2013-01-01-00.00.00.000000, which is the correct result. The result of the second SELECT statement is SQLCODE -180, because a Store Clock value is not valid input to CAST(*string* AS TIMESTAMP).

Actions to take

While in Db2 11 conversion mode, or in Db2 11 new-function mode with application compatibility set to V10R1, identify applications with this incompatibility by starting a trace for IFCID 0366 or IFCID 0376, and then running the applications. Review the trace output for incompatible changes with the identifier 1109. If you need to convert Store Clock values to the TIMESTAMP data type, use the TIMESTAMP built-in function instead of CAST(*string* AS TIMESTAMP).

For example:

```
-- SET APPLICATION COMPATIBILITY TO V11R1
SET CURRENT APPLICATION COMPATIBILITY='V11R1';
-- CONVERT AN 8-BYTE STRING REPRESENTATION OF A STORE CLOCK VALUE
-- TO TIMESTAMP
SELECT TIMESTAMP(X'CAB5060708090100') FROM SYSIBM.SYSDUMMY1;
```

You receive the correct result of 2013-01-01-20.37.04.246928.

Related reference

CAST specification (Db2 SQL)

New maximum lengths for values that are returned for some built-in functions

In Db2 11 new-function mode with application compatibility set to 'V11R1', the maximum lengths for values that are returned for some built-in functions is decreased.

Explanation

For the SPACE and VARCHAR built-in functions, the maximum length of the result is changed from 32767 to 32764 bytes.

Possible impact to your Db2 environment

If the length of the output string for any of these functions is greater than 32764 bytes, SQLCODE -171 is returned.

Actions to take

Review your applications for use of these functions, and, if necessary, modify the function input so that the result does not exceed 32764 bytes. While in conversion mode with application compatibility for your package set to value 'V10R1', run your applications with IFCID 0366 or IFCID 0376 enabled. Then, review the trace output for incompatible changes with the identifier '1110' or '1111'.

Timestamp string representations

String representations of timestamp values must adhere to the rules in the SQL Reference. However, releases before DB2 10 inadvertently tolerate some string representations of timestamps with invalid syntax.

The behavior is controlled by the BIF_COMPATIBILITY subsystem parameter and the application compatibility setting.

- Db2 11 with application compatibility set to V11R1 strictly enforces valid string representations of timestamp values. This is equivalent to DB2 10 with the BIF_COMPATIBILITY subsystem parameter set to CURRENT.
- Db2 11 with application compatibility set to V10R1, the enforcement of valid string representations depends on the BIF_COMPATIBILITY value.

Actions to take

Review your setting of the BIF_COMPATIBILITY subsystem parameter. If the value is not CURRENT, the application compatibility for your package set to V10R1, and you have applications that require string representations of timestamp values supported in a pre-DB2 10, you should make appropriate changes to your SQL to use one of the supported formats.

To modify your applications:

1. Use IFCID 0366 or IFCID 0376 trace to identify applications that depend on the pre-DB2 10 formats.
2. Review the trace output with the function identifier '3' to identify SQL with unsupported timestamp values.
3. Make appropriate changes to your SQL statements.
4. Set the BIF_COMPATIBILITY value to CURRENT.

Related concepts

String representations of datetime values (Db2 SQL)

Application compatibility levels in Db2 (Db2 Application programming and SQL)

Related reference

BIF COMPATIBILITY field (BIF_COMPATIBILITY subsystem parameter) (Db2 Installation and Migration)

Data types of output arguments from a stored procedure call in a Java application

In Db2 11 new-function mode, when a Java application that uses the IBM Data Server Driver for JDBC and SQLJ calls a stored procedure, the data types of stored procedure output arguments match the data types of the parameters in the stored procedure definition.

Explanation

Before DB2 10, if a Java client called a Db2 for z/OS stored procedure, the data types of output arguments matched the data types of the corresponding CALL statement arguments. Starting in DB2 10, the data types of the output arguments match the data types of the parameters in the stored procedure definition.

In Db2 11 conversion mode, or when application compatibility is set to V10R1, you can set the DDF_COMPATIBILITY subsystem parameter to SP_PARMS_JV to keep the behavior that existed before DB2 10. However, when application compatibility is set to V11R1, SP_PARMS_JV is no longer supported.

In Db2 11 with application compatibility set to V11R1, if the version of the IBM Data Server Driver for JDBC and SQLJ is lower than 3.63 or 4.13, a `java.lang.ClassCastException` might be thrown when an output argument value is retrieved.

Actions to take

Take one of the following actions:

- Upgrade the IBM Data Server Driver for JDBC and SQLJ to version 3.63 or 4.13, or later.
- Modify the data types in `CallableStatement.registerOutParameter` method calls to match the parameter data types in the stored procedure definitions. You can set application compatibility to V10R1 and run a trace for IFCID 0366 or 0376 to identify affected applications. Trace records for those applications have a QW0366FN field value of 8.

Related concepts

Application compatibility levels in Db2 (Db2 Application programming and SQL)

CHAR9 and VARCHAR9 functions for compatibility with pre-DB2 10 string formatting of decimal data

DB2 10 changed the formatting of decimal data by the CHAR and VARCHAR built-in functions and CAST specifications with a CHAR or VARCHAR result type. In Db2 11 you can use alternative functions for compatibility with applications that require decimal to string output in the pre-DB2 10 formats:

- CHAR9 (Db2 SQL)
- VARCHAR9 (Db2 SQL)

Important: For portable applications that might run on platforms other than Db2 for z/OS, do not use the CHAR9 and VARCHAR9 functions. Other Db2 family products do not support the these functions.

Actions to take

Review your setting for the BIF_COMPATIBILITY subsystem parameter. If the value is not CURRENT, and you have applications that require decimal to string output in the pre-DB2 10 format, you can rewrite SQL statements to use the CHAR9 and VARCHAR9 functions instead. This approach enables the development of new applications that can accept the current string formatting of decimal data.

To modify your applications to take advantage of the CHAR9, VARCHAR9 functions:

1. Use an IFCID 0366 trace to identify applications that depend on the pre-DB2 10 formats.
2. Rewrite the SQL statements in the identified applications to use the CHAR9 and VARCHAR9 functions instead of the CHAR and VARCHAR functions.
3. Set the BIF_COMPATIBILITY value to CURRENT.

Related reference

BIF COMPATIBILITY field (BIF_COMPATIBILITY subsystem parameter) (Db2 Installation and Migration)

Subsystem parameter BIF_COMPATIBILITY and SQL schemas for compatibility with pre-DB2 10 string formatting of decimal data

DB2 10 changed the formatting of decimal data by the CHAR and VARCHAR built-in functions and CAST specifications with a CHAR or VARCHAR result type. You can temporarily override these changes on a subsystem level by setting the BIF_COMPATIBILITY subsystem parameter to one of the pre-DB2 10 settings. You can also temporarily override these changes on an application level by adding schema SYSCOMPAT_V9 to the front of the PATH bind option or CURRENT PATH special register. The latter approach works for CHAR and VARCHAR functions and does not affect CAST specifications.

The recommended approach is to modify your applications to handle the DB2 10 and later behavior for these functions, as described in the following steps.

Actions to take

To modify your applications to handle the DB2 10 and later behavior for CHAR, VARCHAR, and CAST:

1. Identify applications that need to be modified to handle this change. Run a trace for IFCID 0376 to identify affected applications.
2. Ensure that the BIF_COMPATIBILITY subsystem parameter is set to V9_DECIMAL_VARCHAR.

   To handle the change for the CHAR function only, you can set BIF_COMPATIBILITY to V9, and complete the following steps for the CHAR function.

3. Change any affected applications to handle the DB2 10 and later CHAR and VARCHAR behavior, including stored procedures, non-inline user-defined functions, and trigger packages. Rewrite affected CAST specifications with the appropriate CHAR or VARCHAR function and a CAST to the correct length if needed.
4. Rebind and prepare packages with the PATH(SYSCURRENT,SYSIBM) rebind option. Putting the SYSCURRENT schema at the beginning of the SQL path causes the DB2 10 and later behavior to be used for the CHAR and VARCHAR built-in functions.

   Repeat this step for native stored procedures (SQLPL) and non-inline SQL scalar functions.
5. For views that reference these casts or built-in functions, determine whether the view needs to be changed to have the expected output. Drop and re-create the views with the PATH(SYSCURRENT,SYSIBM) rebind option only if necessary. Rebind any applications that reference the views with the PATH(SYSCURRENT,SYSIBM) option to use the DB2 10 and later CHAR and VARCHAR built-in functions. Repeat this step for inline SQL scalar functions.
6. For materialized query tables or indexes on expressions that reference these casts or built-in functions, drop and re-create the materialized query tables or indexes on expressions with the PATH(SYSCURRENT,SYSIBM) rebind option. Issue the REFRESH TABLE statement for materialized query tables. Rebind any applications that reference the materialized query tables or indexes on expressions with the PATH(SYSCURRENT,SYSIBM) option to use the DB2 10 and later CHAR and VARCHAR built-in functions.
7. Change the value of the BIF_COMPATIBILITY subsystem parameter to CURRENT. When the subsystem parameter value is CURRENT, new applications, rebinds, and CREATE statements use the DB2 10 and later CHAR, VARCHAR, and CAST behavior.

Materialized query tables and expression-based indexes use the CHAR, VARCHAR, and CAST behavior that is specified during its creation. If a reference statement has a different behavior that is specified by the BIF_COMPATIBILITY parameter or a different path, the materialized query table or expression-based index is not used.

Change in CREATE TRIGGER statements with a WHEN clause in which the *search-condition* references a system-period temporal table

If a CREATE TRIGGER statement is issued before Db2 11 new-function mode, and the *triggered-action* contains a WHEN clause in which the *search-condition* references a system-period temporal table, the CREATE TRIGGER statement succeeds, and the trigger continues to work in Db2 11 new-function mode. However, when a CREATE TRIGGER statement with a WHEN clause in which the *search-condition* references a system-period temporal table is issued in Db2 11 new-function mode, the CREATE TRIGGER statement fails. This happens because in Db2 11, the trigger package is created with the SYSTIMESENSITIVE(YES) bind option, which does not allow a reference to the system-period temporal table.

Actions to take

To create a trigger that references a system-period temporal table in the *search-condition* of a WHEN clause, follow these steps:

1. Issue an ALTER TABLE statement with the DROP VERSIONING clause on the system-period temporal table to temporarily disconnect the system-period temporal table from the related history table.
2. Issue the CREATE TRIGGER statement.
3. Issue an ALTER TABLE statement with the ADD VERSIONING clause to redefine the system-period temporal table. In the ALTER TABLE statement, specify the related history table in the USE HISTORY TABLE clause.
4. Issue the REBIND TRIGGER PACKAGE command with option SYSTIMESENSITIVE(NO) on the trigger package that was generated when you performed step “2” on page 59.

Related reference

CREATE TRIGGER (Db2 SQL)

ALTER TABLE (Db2 SQL)
REBIND TRIGGER PACKAGE (DSN) (Db2 Commands)

SQL reserved words

PSPI

Db2 11 introduces several new SQL reserved words, which are listed in Reserved words (Db2 SQL).

In some cases, the use of these reserved words might cause an incompatibility in Db2 11 conversion mode, regardless of the setting of the APPLCOMPAT flag.

Actions to take

Collect IFCID 0366 trace records in Db2 10. Values 4, 5, and 6 for the QW0366FN field indicate instances of reserved words in applications that will cause an incompatibility in Db2 11. Adjust these applications by changing the reserved word to a delimited identifier or by using a word that is not reserved in Db2 11.

PSPI

Built-in function and existing user-defined functions

For built-in and user-defined functions the combination of the function name and the parameter list form the *signature* that Db2 uses to identify the function. If the signatures of new or changed built-in functions in Db2 11 match the signatures existing user-defined functions, applications with unqualified references to the existing user-defined functions might start invoking the new or changed built-in functions instead of the user-defined functions. Db2 11 introduces the following built-in function changes:

GUPI

Table 3. New functions

Function name	Description
ARRAY_AGG (Db2 SQL)	The ARRAY_AGG function returns an array in which each value of the input set is assigned to an element of the array.
ARRAY_DELETE (Db2 SQL)	The ARRAY_DELETE function deletes elements from an array.
ARRAY_FIRST (Db2 SQL)	The ARRAY_FIRST function returns the minimum array index value of an array.
ARRAY_LAST (Db2 SQL)	The ARRAY_LAST function returns the maximum array index value of an array.
ARRAY_NEXT (Db2 SQL)	The ARRAY_NEXT function returns the next larger array index value, relative to a specified array index value.
ARRAY_PRIOR (Db2 SQL)	The ARRAY_PRIOR function returns the next smaller array index value, relative to a specified array index value.
BLOCKING_THREADS (Db2 SQL)	The BLOCKING_THREADS function returns a table that contains one row for each lock or claim that threads hold against specified databases.
CARDINALITY (Db2 SQL)	The CARDINALITY function returns the number of elements in an array.

Table 3. New functions (continued)

Function name	Description
CHAR9 (Db2 SQL)	The CHAR9 function returns a fixed-length character string representation of the argument. The CHAR9 function is intended for compatibility with previous releases of Db2 for z/OS that depend on the result format that is returned for decimal input values in Version 9 and earlier. **Important:** For portable applications that might run on platforms other than Db2 for z/OS, use the CHAR function instead. Other Db2 family products do not support the CHAR9 function.
MAX_CARDINALITY (Db2 SQL)	The MAX_CARDINALITY function returns the maximum number of elements that an array can contain.
MEDIAN	The MEDIAN function returns the median of a set of numbers. This function can run only on an accelerator server.
TRIM_ARRAY (Db2 SQL)	The TRIM_ARRAY function deletes elements from the end of an ordinary array.
VARCHAR9 (Db2 SQL)	The VARCHAR9 function returns a fixed-length character string representation of the argument. The VARCHAR9 function is intended for compatibility with previous releases of Db2 for z/OS that depend on the result format that is returned for decimal input values in Version 9 and earlier. **Important:** For portable applications that might run on platforms other than Db2 for z/OS, use the VARCHAR function instead. Other Db2 family products do not support the VARCHAR9 function.

GUPI

Actions to take

To continue to execute a user-defined function with the same name or signature as a new built-in function or signature, qualify the name of the existing user defined function in your application. For more information about Db2 resolves qualified and unqualified references to functions, see Function resolution (Db2 SQL).

SQLCODE changes

Some SQLCODE numbers and message text might have changed in Db2 11. Also, the conditions under which some SQLCODEs are issued might have changed. For more information, see New, changed, and deleted codes (Db2 Codes).

GUPI

V10R1 application compatibility level

When you set the application compatibility level to V10R1, applications that attempt to use functions and features that are introduced in Db2 11 or later might behave differently or receive an error.

When your Db2 11 environment is migrated to new-function mode, you can run individual applications with some of the features and behavior of DB2 10. Your applications can continue to experience V10R1 behavior while in Db2 11 new-function mode. Then, you can migrate each application to a new

application compatibility value separately until all are migrated. If application compatibility is set to V10R1 and you attempt to use the new functions of a later version, SQL might behave differently or result in a negative SQLCODE, such as SQLCODE -4743.

PSPI

You can run package level accounting or monitor traces with IFCID 0239 and review field QPACINCOMPAT, which indicates an SQL incompatible change. If a trace is started for IFCID 0366 or IFCID 0376 and application compatibility is set for a previous version, details about features and functions that have a change in behavior are written in field QW0366FN or QW0376FN.

PSPI

A migrated Db2 11 environment in conversion mode behaves with V10R1 application compatibility. Application and SQL incompatibilities are described in the migration information for each version.

The following table shows many of the features and functions that are controlled by application compatibility, and the results if you specify V10R1. If a behavior difference is traced, then the IFCID trace function code is shown.

Table 4. Behavior of V10R1 application compatibility

Feature or Function	Result with V10R1 application compatibility	IFCID 0366 or IFCID 0376 trace function code
An SQL statement in a client application includes an unsupported conversion (from a string type to a numeric type or from a numeric type to a string type), and implicit casting is disabled (DDF_COMPATIBILITY is set to SP_PARMS_NJV or to DISABLE_IMPCAST_NJV).	SQLCODE -301	7"1" on page 64
A client application executes an SQL CALL statement to execute a Db2 for z/OS stored procedure. The DDF_COMPATIBILITY subsystem parameter is set to SP_PARMS_NJV for client applications other than Java applications, or SP_PARMS_JV for Java applications.	The data types of the data that is returned from the SQL CALL statement match the data types of the CALL statement arguments. This behavior is compatible with the behavior before Version 10.	8"1" on page 64
A client application accesses Db2 11 from an IBM Data Server Driver for JDBC and SQLJ client. The DDF_COMPATIBILITY subsystem parameter is set to IGNORE_TZ for Java applications.	The Db2 server ignores the TIMEZONE portion, appended by the IBM Data Server Driver for JDBC and SQLJ, of the value in the TIMESTAMP WTIH TIMEZONE input to a TIMESTAMP target. This behavior is compatible with the behavior before DB2 10.	9
BIF_COMPATIBILITY is set to V9_TRIM, and input *string-expression* is EBCDIC mixed data for the RTRIM, LTRIM, or STRIP built-in function.	The DB2 9 version of SYSIBM.LTRIM(*string-expression*), SYSIBM.RTRIM(*string-expression*), or SYSIBM.STRIP(*string-expression*) is executed.	10
An implicit insert or update of an XML document node	SQLCODE -20345	1101

Table 4. Behavior of V10R1 application compatibility (continued)

Feature or Function	Result with V10R1 application compatibility	IFCID 0366 or IFCID 0376 trace function code
A predicate expression with an explicit cast or an operation with an invalid value that does not affect the results of XPath processing	SQLCODE -20345	1102
How the resource limit facility uses ASUTIME value for nested routines	SQLCODE -905 is issued only when the ASUTIME limit of the top-level calling package is encountered.	1103
The lengths of values that are returned from CURRENT CLIENT_USERID, CURRENT CLIENT_WRKSTNNAME, CURRENT CLIENT_APPLNAME, or CURRENT CLIENT_ACCTNG special register are longer than the DB2 10 limits.	The special register values are truncated to the DB2 10 maximum lengths and padded with blanks	1104, 1105, 1106, 1107
A CAST(string as TIMESTAMP) specification with an input string of length of 8 or an input string of length 13	An explicit cast specification from string as TIMESTAMP interprets an 8-byte character string as a Store Clock value and a 13-byte string as a GENERATE_UNIQUE value. CAST result might be incorrect.	1109
Invocation of the SPACE or VARCHAR built-in function when the result is defined as VARCHAR(32765), VARCHAR(32766), or VARCHAR(32767)	No error	1110, 1111
Subsystem parameter XML_RESTRICT_EMPTY_TAG is set to YES, and an empty XML element is serialized as <*emptyElement*></*emptyElement*>	No error	1112
Specification of bind option DBPROTOCOL(DRDACBF)	DSNT298I	
A period specification that follows the name of a view in the FROM clause of a query	SQLCODE -4743	
A period clause that follows the name of a target view in an UPDATE or DELETE statement	SQLCODE -4743	
A SET CURRENT TEMPORAL SYSTEM_TIME statement	SQLCODE -4743	
A SET CURRENT TEMPORAL BUSINESS_TIME statement	SQLCODE -4743	
A reference to a global variable	SQLCODE -4743	

Table 4. Behavior of V10R1 application compatibility (continued)

Feature or Function	Result with V10R1 application compatibility	IFCID 0366 or IFCID 0376 trace function code
Use of array operations and built-in functions such as • Use of the UNNEST collection-derived-table • Use of the ARRAY_FIRST, ARRAY_LAST, ARRAY_NEXT, ARRAY_PRIOR, ARRAY_AGG, TRIM_ARRAY, CARDINALITY, MAX_CARDINALITY built-in functions • A SET assignment-statement of an array element as a target table • A CAST specification with a parameter marker as the source and an array as the data type	SQLCODE -4743	
An aggregate function that contains the keyword DISTINCT and references a column that is defined with a column mask	SQLCODE -20478	
An SQL statement contains the GROUP BY clause and references a column that is defined with a column mask	SQLCODE -20478	
An SQL statement contains the set operator UNION ALL or UNION DISTINCT and references a column that is defined with a column mask	SQLCODE -20478	
A reference to an alias for a sequence object	SQLCODE -4743	
A reference to an unqualified sequence that is not resolved to a public alias	SQLCODE -204	
A SELECT with a table function reference that includes a typed correlation clause	SQLCODE -4743	
A CALL statement that specifies an autonomous procedure	SQLCODE -4743	
The following datetime assignments: • A valid string representation of a timestamp to a date column • A valid string representation of a timestamp to a time column • A valid string representation of a date to a timestamp column	SQLCODE -180	

Notes:

1. PSPI To find details about the incompatible parameters, examine the contents of fields QW0366SC_Var, QW0366PR_Var, and QW0366INC_Var, or QW0376SC_Var, QW0376PR_Var, and QW0376INC_Var. See the *prefix*.SDSNIVPD(DSNWMSGS) file for more information. PSPI

Utility release incompatibilities

When you migrate to Db2 11, be aware of the utility release incompatibilities.

Plan for the following changes in Db2 11 that might affect your migration.

Release incompatibilities that were changed or added since the first edition of this Db2 11 publication are indicated by a vertical bar in the left margin. In other areas of this publication, a vertical bar in the margin indicates a change or addition that has occurred since the Db2 10 release of this publication.

Changes to the mapping table for REORG

Explanation

In Db2 11, the format of mapping tables has changed to accommodate the new 10-byte format of the RBA and LRSN values. (Mapping tables are used by the REORG TABLESPACE utility when SHRLEVEL CHANGE is specified.) The LRSN column in the mapping table has changed to CHAR(10) instead of CHAR(6).

Possible impact to your Db2 environment

Beginning in Db2 11 new-function mode, if you use the DB2 10 format for a mapping table and try to specify it in your REORG statement, REORG does not use it. Instead, the utility implicitly creates a mapping table in the Db2 11 format in the same database as the mapping table that you specified.

Actions to take

Either create your mapping tables according to the Db2 11 format or take advantage of the new feature where REORG creates the mapping tables for you.

Related concepts

Automated REORG mapping table management
In Db2 11, you no longer have to create your own mapping table and index before running the REORG TABLESPACE utility. If you do not specify a mapping table in your utility statement, REORG creates one for you.

Before running REORG TABLESPACE (Db2 Utilities)

Differences in page set format impacting DSN1COPY utility

Explanation

In Db2 11 new-function mode (NFM), you can convert page sets from basic format to extended format to support 10 byte log RBA and LRSN values. Such extended format page sets are not compatible with DB2 10 or earlier releases. The use of DSN1COPY to copy an extended format page set to a Db2 system running on a prior release of Db2 results in errors during subsequent access on that system.

Possible impact to your Db2 environment

Errors might occur on the Db2 system to which the extended format page set is copied if that system is running a release of Db2 prior to Db2 11 or if the system is running Db2 11 in conversion mode.

Actions to take

Prevent DSN1COPY or other moves of extended format page sets from a Db2 11 system to a system running a prior release of Db2. If DSN1COPY is used to copy page sets between systems, do not convert page sets to extended format until all relevant systems have been migrated to Db2 11.

Parallelism change for the COPY and RECOVER utilities

Explanation

In Db2 11 conversion mode, the degree of parallelism can decrease for the COPY and RECOVER utilities.

The new PARAMDEG_UTIL subsystem parameter now applies to COPY and RECOVER control statements that include the PARALLEL keyword.

If you specify the PARALLEL keyword, the number of subtasks for PARALLEL always overrides the specification of the PARAMDEG_UTIL subsystem parameter. Therefore, PARALLEL can be smaller or larger than the value of PARAMDEG_UTIL.

Possible impact to your Db2 environment

The maximum number of objects that are processed in parallel for the COPY and RECOVER utilities might be lower than in previous releases.

Actions to take

If you want to increase the number of objects that are processed in parallel, increase the value of the PARAMDEG_UTIL subsystem parameter.

Parallelism change for the CHECK INDEX, LOAD, REBUILD INDEX, and REORG TABLESPACE utilities

Explanation

In Db2 11, the following utilities can use larger degrees of parallelism:

- REORG TABLESPACE
- REBUILD INDEX
- CHECK INDEX
- LOAD

The main goal of the increased degree of parallelism is to reduce elapsed times. However, extra virtual storage might be needed. Sort operations that are invoked by the utilities are the main consumers of the storage. If the amount of available virtual storage is not sufficient for the extra demand, the increased parallelism might drive CPU consumption and elapsed times higher.

The region size of the utility job (as set by the REGION parameter of the job, or by an IEFUSI exit) controls the amount of virtual storage that might be used. The region contains the administrative portion of the sort work and the buffers, which are sorted in each subtask. This virtual storage requirement is usually in the range of a few megabytes per task, but it can grow to a few hundred megabytes for large sorts (in the terabyte range). This memory consumption is essential for good performance. In most cases, it does not have a large effect on real storage.

In addition, sort products, such as Db2 SORT and DFSORT, can use auxiliary storage to store intermediate results. The auxiliary storage is used as an alternative to writing the pre-sorted pieces to sort work data sets. The auxiliary storage use can easily grow into the multiple gigabyte range for large sorts.

Possible impact to your Db2 environment

The increased parallelism can reduce elapsed times. However, because more storage might be required, you must consider the trade-offs. If virtual storage for the region is constrained, CPU time and elapsed time might increase for large sort operations because the sort operations become inefficient. If real storage is constrained and more auxiliary storage is used by the extra, the increased paging might impact system performance.

The amount of extra storage that is used depends on the size of the data to be sorted, the sort product that is used, and the configuration of the sort product.

Actions to take

To take advantage of the increased parallelism and achieve improved elapsed times:

- Use the REGION parameter value of the utility job (or IEFUSI exit setting) to increase the amount of virtual storage that can be used. DFSORT invocations now use the all of the available storage in the region size. Db2 Sort for z/OS has always used all of the available storage in the region size.
- Configure the sort products to write intermediate results to work files, instead of auxiliary storage, if real storage consumption is a problem.

If your system cannot tolerate the increased virtual storage and CPU consumption:

- Specify the PARALLEL *n* keyword in the utility control statement, or the value of the PARAMDEG_UTIL subsystem parameter, to restrict the degree of parallelism.
- Use the REGION parameter value (or IEFUSI exit setting) to limit the amount of storage that is used.
- Configure the sort products to write intermediate results to work files, instead of auxiliary storage, if real storage is a problem.

Related reference

MAX UTILS PARALLELISM field (PARAMDEG_UTIL subsystem parameter) (Db2 Installation and Migration)

REGION parameter (MVS JCL Reference)

Preparing to customize Db2 Sort for z/OS (Db2 Sort for z/OS)

Related information

DFSORT Installation and Customization

Changes to REORG default values

Explanation

In Db2 11 conversion mode, the following changes are made to the default values for the REORG utilities:

The following changes are made to the default values for the REORG TABLESPACE utility:

- The default DRAIN value is changed from WRITERS to ALL.
- The NOPAD keyword is now the default value in the UNLOAD EXTERNAL clause and the DISCARD clause.

The following changes are made to the default values for the REORG INDEX utility:

- The default DRAIN value is changed from WRITERS to ALL.

Possible impact to your Db2 environment

By default, Db2 drains all readers and writers during the log phase, instead of only writers.

By default, the variable-length columns in the unloaded or discarded records are to occupy the actual data length without additional padding.

Actions to take

If you want Db2 to drain only the writers during the log phase, specify DRAIN WRITERS in your utility control statement.

If you want the REORG processing to pad variable-length columns, specify NOPAD NO in your utility control statement.

Change to DSNU126I return code when running REORG on an LOB table space

Explanation

In Db2 11 conversion mode, the return code from message DSNU126I changed to 8.

Running REORG TABLESPACE SHRLEVEL NONE on an LOB table space was deprecated in a prior release and completed with return code 0. The return code of message DSNU126I has changed to return code 8 to indicate the function is no longer supported.

Possible impact to your Db2 environment

REORG jobs that previously completed with message DSNU126I return code 0 might now fail with DSNU126I return code 8.

Actions to take

Review your REORG jobs for instances of DSNU126I and remove any REORG TABLESPACE SHRLEVEL NONE on a LOB table space.

Changes to RECOVER utility

Explanation

In Db2 11 conversion mode, the following changes are made to the RECOVER utility:

- TOLOGPOINT, TORBA, and RESTOREBEFORE keywords can accept basic 6-byte format or extended 10-byte format based on the length of the RBA or LRSN value that is specified. Previously, any length was accepted and then extended or truncated as required.

Possible impact to your Db2 environment

Operands of 6 bytes or less are interpreted as being in basic 6-byte format. Operands greater than 6 bytes are interpreted as being in extended 10-byte format. In both cases, padding on the left with X'00' occurs to form complete 6-byte or 10-byte operands. Conversion between basic and extended format is performed as required for the recovery operation.

Actions to take

Update the RBA or LRSN values that are specified for the TOLOGPOINT, TORBA, or RESTOREBEFORE keywords to indicate either basic 6-byte format or extended 10-byte format.

Changes to DSNACCOX stored procedure default values

In Db2 11 new-function mode (NFM), the following changes are made to the defaults of the DSNACCOX stored procedure:

Explanation

RRTDataSpaceRat input parameter default value is -1. Previously, it was 2.0.

Possible impact to your Db2 environment

Review your calls to the DSNACCOX stored procedure. Look for NULL as the value of *RRTDataSpaceRat*. The new default turns off this criterion. Any positive values continue to be processed as in DB2 10.

Actions to take

To continue the DB2 10 behavior, change *RRTDataSpaceRat* input parameter values of NULL to 2.0 to match the DB2 10 default. To turn off this criterion, leave the default as NULL.

Related reference

DSNACCOX stored procedure (Db2 SQL)

Changes to DSNACCOX stored procedure result set

In Db2 11 new-function mode (NFM), new values are possible in the OBJECTTYPE column of the DSNACCOX recommendation result set.

Explanation

'XS' for XML table spaces and 'LS' for LOB table spaces are now possible values of the OBJECTTYPE column results.

Possible impact to your Db2 environment

Review your processing of results from the DSNACCOX stored procedure. Unexpected values might be handled as invalid.

Actions to take

Update your process to handle the new possible values.

Related reference

DSNACCOX stored procedure (Db2 SQL)

Changes to DSNACCOX stored procedure processing for REORG and COPY recommendations

In Db2 11 new-function mode (NFM), more information is evaluated when REORG or COPY is recommended.

Explanation

When the input parameter *QueryType* specifies REORG or COPY recommendations, DSNACCOX also checks the database exception table (DBET) entry for an exception state.

Possible impact to your Db2 environment

Review your processing of results from the DSNACCOX stored procedure. Unexpected values might be handled as invalid. Database exception table (DBET) states are added to the OBJECTSTATUS column of the result set.

Actions to take

Update your process to handle the new possible values.

Related reference

DSNACCOX stored procedure (Db2 SQL)

Changes to DSNACCOX stored procedure processing for ChkLvl 8

In Db2 11 new-function mode (NFM), a new row is not inserted if the result set already has a recommendation for a utility operation.

Explanation

When the input parameter *ChkLvl* specifies level 8 processing, DSNACCOX adds the utility operation recommendation to an existing row if one exists for the object. If an existing row does not exist, DSNACCOX continues to add a row.

Possible impact to your Db2 environment

Review your processing of results from the DSNACCOX stored procedure. Unexpected values might be handled as invalid. New rows for objects are only added to the result set if the object is not already present.

Actions to take

Update your process to handle the new possible values.

Related reference

DSNACCOX stored procedure (Db2 SQL)

ALTER statements that change limit keys are pending changes

Starting in Db2 11 new function mode, ALTER statements that modify limit key values result in pending data definition changes, which do not take effect until materialized by the REORG utility. They also block all subsequent immediate changes until materialized.

Explanation

In Db2 11, ALTER statements that change limit keys for the following types of table spaces result in pending data definition changes:

- partition-by-range spaces
- Partitioned (non-UTS) tables spaces with table-controlled partitioning

Affected partitions are placed in advisory REORG-pending (AREOR) status. In DB2 10, such ALTER statements resulted in immediate changes, which placed affected partitions in restrictive REORG-pending (REORP) status.

ALTER statements that alter the last partition are an exception if they change MAXVALUE to a value less than MAXVALUE for ascending, or from MINVALUE to a value greater than MINVALUE for descending. In such cases, the changes are immediate and affected partitions are placed in restricted REORG-pending status.

Possible impact to your Db2 environment

Affected partitions remain in advisory REORG-pending (AREOR) status, the old limit key values remain in effect, and the data remains available until the pending definition change is materialized. However, all subsequent immediate data definition changes, including in the same statement, remain blocked until the pending data definition changes are materialized. Table spaces with pending definition changes have an entry in SYSIBM.SYSPENDINGDDL.

Also, you can no longer materialize changes for ALTER statements that change limit keys by using the REORG TABLESPACE utility with SHRLEVEL NONE or the LOAD utility with REPLACE.

Actions to take

Modify any existing jobs that materialize limit key changes to run the REORG TABLESPACE utility with SHRLEVEL CHANGE or SHRLEVEL REFERENCE.

For jobs that use the LOAD utility with REPLACE, modify the job to run the REORG TABLESPACE utility with SHRLEVEL CHANGE or SHRLEVEL REFERENCE before running LOAD.

Related concepts

Improved data availability when altering limit keys
In Db2 11, you can alter the limit keys for partitioned table spaces without impacting the availability of the data.

Pending data definition changes (Db2 Administration Guide)

Related tasks

Materializing pending definition changes (Db2 Administration Guide)

Altering table spaces (Db2 Administration Guide)

Related reference

REORG-pending status (Db2 Utilities)

Command release incompatibilities

When you migrate from Db2 10 to Db2 11, be aware of the command release incompatibilities.

GUPI

Plan for the following changes in Db2 11 that might affect your migration.

Release incompatibilities that were changed or added since the first edition of this Db2 11 publication are indicated by a vertical bar in the left margin. In other areas of this publication, a vertical bar in the margin indicates a change or addition that has occurred since the Db2 10 release of this publication.

Change to DISPLAY UTILITY output

Explanation

The output for the DISPLAY UTILITY command now includes the date and the time when the job was submitted.

Possible impact to your Db2 environment

Applications that parse DISPLAY UTILITY output might need to be updated.

Actions to take

Determine if any of your applications parse output of the DISPLAY UTILITY command and update the applications if needed.

GUPI

Removing the SYSPUBLIC schema from the PATH bind option

Starting in Db2 11 conversion mode, SYSPUBLIC is the schema that is used for public aliases. As such, the PATH bind option must not specify the SYSPUBLIC schema.

Explanation

In previous versions, you were not restricted from specifying SYSPUBLIC as part of the PATH bind option. With Db2 11 you will no longer be able to specify SYSPUBLIC as part of the PATH bind option.

Possible impact to your Db2 environment

Creation or resolution of some objects that worked in previous versions of Db2 might fail in Db2 11 with SQLCODE -713 if SYSPUBLIC is specified as part of the PATH bind option.

Actions to take

Query the catalog to see if any object schemas use SYSPUBLIC as the schema qualifier. This is highly unlikely for any object, but most likely with objects that use the PATH (functions, procedures, and sequences).

Change any existing PATH bind option to not specify SYSPUBLIC as a schema.

Related reference

PATH bind option (Db2 Commands)

Related information

-713 (Db2 Codes)

LIBRARY bind option delimiters

Starting in Db2 11, you might need to add single quotation mark (') delimiters to LIBRARY bind option values that specify partitioned data sets.

Explanation

Before Db2 11, LIBRARY bind option values that specify partitioned data set names can be specified with or without single quotation mark delimiters. Partitioned data set names without delimiters are taken as is if the bind command has no TSO segment. When the bind command has a TSO segment, Db2 prefixes

nondelimited partitioned data set names with a user name. Db2 11 introduces support for zFS DBRM files. A result is that when the bind command has no TSO segment, the partitioned data set name is also prefixed with a default user name. The timing for resolving the dataset name is changed such that Db2 has no information about whether the bind command has a TSO segment.

Possible impact to your Db2 environment

BIND PACKAGE commands with a LIBRARY option value that indicates a non-delimited value for partitioned data set might fail with z/OS TSO/E message IKJ56228I.

Actions to take

Determine whether BIND PACKAGE commands must be updated. If a BIND command fails with the z/OS TSO/E message IKJ56228I, add single quotations mark delimiters to the dataset name. If a BIND package command runs successfully, no action is required.

Related reference

LIBRARY bind option (Db2 Commands)

Storage release incompatibilities

When you migrate to Db2 11, be aware of the storage release incompatibilities.

Plan for the following changes in Db2 11 that might affect your migration.

Release incompatibilities that were changed or added since the first edition of this Db2 11 publication are indicated by a vertical bar in the left margin. In other areas of this publication, a vertical bar in the margin indicates a change or addition that has occurred since the Db2 10 release of this publication.

New minimum for HVSHARE

Explanation

In Db2 11, the required amount of contiguous 64-bit shared private storage for each Db2 subsystem is 1 TB. In previous releases, the minimum requirement was 128 GB.

Possible impact to your Db2 environment

If you do not have an adequate amount of contiguous 64-bit shared private storage, Db2 11 will not start.

Actions to take

Before migrating to Db2 11, ensure that you have at least 1 TB of contiguous 64-bit shared private storage for each Db2 subsystem.

Related concepts

Shared memory storage requirements (Db2 Installation and Migration)

Other release incompatibilities

When you migrate to Db2 11, be aware of the release incompatibilities.

Plan for the following changes in Db2 11 that might affect your migration.

Release incompatibilities that were changed or added since the first edition of this Db2 11 publication are indicated by a vertical bar in the left margin. In other areas of this publication, a vertical bar in the margin indicates a change or addition that has occurred since the Db2 10 release of this publication.

Changed default values for some subsystem parameters and installation panel fields

Explanation

The default values for some subsystem parameters and installation panel fields have changed. The new values are listed in the table below.

Table 5. Subsystem parameters and installation panel fields with new default values

Parameter	Field	Panel	Db2 10 default value	Db2 11 default value
REORG PART SORT NPSI field (REORG_PART_SORT_NPSI subsystem parameter) (Db2 Installation and Migration)	REORG PART SORT NPSI (none in Db2 10)	DSNTIP62	NO	AUTO
SUBQ_MIDX	none	none	DISABLE	ENABLE

Possible impact to your Db2 environment

This change has no impact on your Db2 environment if you maintain your current subsystem parameter settings during migration.

Actions to take

If your Db2 subsystem uses the Db2 10 default value for any of these parameters, review the parameter descriptions to determine if the Db2 11 default value is more appropriate.

Changed maximum values for some subsystem parameters

Explanation

The maximum values for some subsystem parameters have changed. The new values are listed in the table below.

Table 6. Subsystem parameters that have new maximum values

Subsystem parameter	Db2 10 maximum value	Db2 11 maximum value
EDM SKELETON POOL SIZE field (EDM_SKELETON_POOL subsystem parameter) (Db2 Installation and Migration)	2097152	4194304
EDM DBD CACHE field (EDMDBDC subsystem parameter) (Db2 Installation and Migration)	2097152	4194304
EDM STATEMENT CACHE field (EDMSTMTC subsystem parameter) (Db2 Installation and Migration)	1048576	4194304
MAX KEPT DYN STMTS field (MAXKEEPD subsystem parameter) (Db2 Installation and Migration)	65535	204800

Actions to take

If your subsystems operate at or near the old maximum value, review the new maximum values and determine if your subsystem would benefit from increasing the value of any of these subsystem parameters.

Change to job DSNTEJ6Z

Explanation

In Db2 11, the DSN8ED7 installation verification program calls the SYSPROC.ADMIN_INFO_SYSPARM stored procedure instead of the SYSPROC.DSNWZP stored procedure.

Possible impact to your Db2 environment

If you do not have the ADMIN_INFO_SYSPARM stored procedure installed the installation verification procedure fails.

Actions to take

If you run job DSNTEJ6Z, which prepares and runs the DSN8ED7 program, ensure that the ADMIN_INFO_SYSPARM stored procedure is installed and correctly configured.

Change to log capture exit routine

Explanation

PSPI

In Db2 11, log buffers are moved from 31-bit key 7 private storage to 64-bit key 0 fetch-protected common storage. Therefore, the log capture exit routine must be invoked in 64-bit mode. In Db2 10, the log capture exit routine is called in AMODE(31), key 7. In Db2 11, the log capture exit routine is called in AMODE(64), key 0. Also in Db2 10, the parameter list consists of two 31-bit pointers. The parameter list now contains two 64-bit pointers that point to the standard EXPL parameter list and to the log capture exit parameter list (LOGX). As in prior releases, the log capture exit routine can be invoked in either TCB or SRB mode.

Possible impact to your Db2 environment

If the log capture exit is not updated to handle the new environment and the modified parameter list, an abend might occur. If you use a data replication product that uses the log capture exit, that product will not work until it is updated.

Actions to take

If you use the log capture exit routine, ensure that the exit is updated to handle the new environment and the modified parameter list.

PSPI

Expanded fields in trace records

Explanation

PSPI

The following changes are made to trace records in Db2 11:

- Trace record fields that contain RBA or LRSN values expand to 10 bytes. In most cases, the expanded RBA or LRSN fields move so that offsets of fields that follow do not change. However, field QW0204UR cannot move.
- Some fields in IFCID 0002 and IFCID 0225 that are related to shareable storage have been expanded from 4 to 8 bytes, moved, and renamed.

Possible impact to your Db2 environment

These changes might affect your environment in the following ways:

- Programs that format IFCID 0204 output might generate incorrect information.
- Programs that refer to IFCID 0002 and IFCID 0225 fields by name might need to be changed.

Actions to take

Adjust your programs as follows:

- Adjust programs that format trace information to account for the increased size of the QW0204UR field, and the modified offsets of the fields that follow it.
- Adjust programs that refer to IFCID 0002 and IFCID 0225 in the following ways:
  - References to field QISEKSPA in IFCID 0002 must be changed to use new field QISEKSPA8.
  - References to field QW0225SC in IFCID 0225 must be changed to use new field QW0225SC8.
  - References to field QW0225LS in IFCID 0225 must be changed to use new field QW0225LS8.
  - References to field QW0225SX in IFCID 0225 must be changed to use new field QW0225SX8.
  - References to field QW0225HS in IFCID 0225 must be changed to use new field QW0225HS8.

PSPI

Moved fields in trace records

Explanation

PSPI

The following changes are made to trace records in Db2 11:

- Field QW0225DMH in IFCID 0225 has moved. In Db2 10, this field follows field QW0225F2. In Db2 11, this field is at the end of section 4, following field QW0225HS8.

Possible impact to your Db2 environment

This change might affect your environment in the following ways:

- Programs that refer to IFCID 0225 fields by offset need to be changed.

Actions to take

Adjust your programs as follows:

- Ensure that your applications use the latest DSNDQW03 mapping macro, so that the applications reference
- the correct IFCID 0225 fields.

PSPI

Changed default classes for monitor trace

Explanation

PSPI

In previous releases of Db2, when a monitor trace was started without the CLASS keyword, only monitor trace class 1 was started. In Db2 11, monitor trace class 11 is also started.

Possible impact to your Db2 environment

The return area that is defined in your IFI applications might be too small to hold the additional trace records.

Actions to take

Adjust your programs in one of the following ways:

- Specify the CLASS or IFCID keyword in your START TRACE command so that you start only the trace classes or IFCIDs that you need.
- Increase the size of the IFI return area.

PSPI

Client information in messages

In Db2 11 conversion mode, the length of client information values in several messages might change significantly. The maximum length and blank padding of client information values for user ID, workstation name, application name, and accounting information as set by the connection.

Explanation
In Db2 11 the following messages length changes for client information values:

- DSNB260I
- DSNI031I
- DSNJ031I
- DSNR035I
- DSNR048I
- DSNT318I
- DSNT375I
- DSNT376I
- DSNT377I
- DSNT378I
- DSNV436I
- DSNV437I

Possible impact to your Db2 environment

Any programs that read message text must tolerate changes to the length of token values.

Actions to take

If you have applications that read the message text, change your applications to process the updated text.

The SPUFI DBRM (DSNESM68) is now versioned

Explanation
In past releases, DSNESM68, the DBRM for the SPUFI facility, was not versioned. It also had a fixed consistency token of x'149EEA901A79FE48', making it possible for SPUFI to connect to remote locations where the package was bound only locally. Although this provision was convenient, it raised the risk of enhancements and service changes. In Db2 11, DSNESM68 is versioned.

Possible impact to your Db2 environment
This change might affect how you must bind SPUFI to connect to remote locations.

Actions to take
Use standard procedure to bind SPUFI at a remote location using a versioned DSNESM68 DBRM. The following example shows how to bind Db2 11 SPUFI to connect to a remote DB2 10 server. Use the bind

statements for the DSNESPCS package and plan as a model for binding the DSNESPRR and DSNESMUR packages and plans.

```
DSN SYSTEM(v11_ssid)
  BIND PACKAGE(DSNTIAP) MEMBER(DSNTIAP) -
       ACTION(ADD) ENCODING(EBCDIC) -
       LIBRARY('db2_v11_prefix.SDSNDBRM')
  BIND PACKAGE(v10_location.DSNTIAP) MEMBER(DSNTIAP) -
       ACTION(ADD) ENCODING(EBCDIC) -
       LIBRARY('db2_v11_prefix.SDSNDBRM')

  BIND PACKAGE(DSNESPCS) MEMBER(DSNESM68)
       ACTION(ADD) ISO(CS) CURRENTDATA(YES) ENCODING(EBCDIC) -
       LIBRARY('db2_v11_prefix.SDSNDBRM')
  BIND PACKAGE(v10_location.DSNESPCS) MEMBER(DSNESM68)
       ACTION(ADD) ISO(CS) CURRENTDATA(YES) ENCODING(EBCDIC) -
       LIBRARY('db2_v11_prefix.SDSNDBRM')
  BIND PLAN(DSNESPCS) -
       PKLIST(*.DSNESPCS.DSNESM68, -
              *.DSNTIAP.DSNTIAP) -
       ACTION(ADD) RETAIN -
       ISO(CS) CURRENTDATA(YES) ENCODING(EBCDIC)
```

New impact of IIPHONORPRIORITY parameter

Explanation
In Db2 11 , when the IIPHONORPRIORITY parameter in the IEAOPT*xx* member of SYS1.PARMLIB is set to NO, Db2 does not allow system tasks to become eligible for IBM Z Integrated Information Processor (zIIP) processing.

Possible impact to your Db2 environment
If your IIPHONORPRIORITY parameter is set to NO, you might observe fewer tasks being routed to zIIP specialty engines in Db2 11 .

Actions to take
Examine your zIIP workloads and determine whether your current IIPHONORPRIORITY setting is still appropriate for your environment.

- If IIPHONORPRIORITY is changed dynamically to YES, restart Db2 to allow system tasks to become eligible for zIIP processing.
- If IIPHONORPRIORITY is changed dynamically to NO, restart Db2 to stop system tasks being routed to zIIP specialty engines.

Changed messages and codes

The text of some messages and codes is changed. In general, the change is effective in conversion mode. If you have applications that read the message and code text, change your applications to process the updated text. See the troubleshooting information for lists of changed messages and codes. Your application must also tolerate changes to the length of token values.

Related reference
Changed messages (Db2 Messages)
Changed SQL codes (Db2 Codes)

Chapter 9. Command changes in Db2 11

Use this information to plan for changes to commands in Db2 11.

New commands in Db2 11

GUPI

Table 7. New commands

Command	Description
BIND SERVICE (DSN) (Db2 Commands)	The BIND SERVICE subcommand builds an application package that represents a Db2 REST service.
-DISPLAY BLOCKERS (Db2) (Db2 Commands)	The DISPLAY BLOCKERS command displays locks and claims that active threads hold against specified databases.
FREE SERVICE (DSN) (Db2 Commands)	The FREE SERVICE subcommand deletes an application package that represents a Db2 REST service.
-START CDDS (Db2) (Db2 Commands)	The START CDDS command directs all members of a data sharing group to allocate and open the compression dictionary data set (CDDS).
-STOP CDDS (Db2) (Db2 Commands)	The STOP CDDS command directs all members of a data sharing group to close and deallocate the CDDS.
-DISPLAY RESTSVC (Db2) (Db2 Commands)	The DISPLAY RESTSVC command displays the status of REST services that exist in Db2.
-START RESTSVC (Db2) (Db2 Commands)	The START RESTSVC command activates the definition of a REST service that is stopped.
-STOP RESTSVC (Db2) (Db2 Commands)	The STOP RESTSVC command prevents Db2 from accepting any new discover details or invoke requests for one or more REST services. .

Changed commands in Db2 11

The following table shows the existing commands that have new and changed options. For information about an option with a changed default value, see Command release incompatibilities.

GUPI

Table 8. Changes to existing commands

Command	Description of enhancements and notes
-ACCESS DATABASE (Db2)	Additional option values: • MODE (STATS)

© Copyright IBM Corp. 2013, 2019

Table 8. Changes to existing commands (continued)

Command	Description of enhancements and notes
-ALTER BUFFERPOOL (Db2)	Additional options: • FRAMESIZE(4K|1M|2G) • SPSIZE(*integer*) • SPSEQT(*integer*) • VPSIZEMAX(*|*integer*) • VPSIZEMIN(*|*integer*) **PGFIX and FRAMESIZE at migration:** Buffer pools that are set to PGFIX(YES) during migration to Db2 11 can use FRAMESIZE(1M) by default in Db2 11. However, buffer pools that are set to PGFIX(NO) during migration use 4 KB frames in Db2 11 even if altered to PGFIX(YES), unless you explicitly specify a different FRAMESIZE value.
-ALTER GROUPBUFFERPOOL (Db2)	Additional option values: • CLASST(*class-threshold1*,*class-threshold2*)
BIND PACKAGE and REBIND PACKAGE: APPLCOMPAT bind option (Db2 Commands) APREUSE bind option (Db2 Commands) ARCHIVESENSITIVE bind option (Db2 Commands) BUSTIMESENSITIVE bind option (Db2 Commands) DBPROTOCOL bind option (Db2 Commands) DESCSTAT bind option (Db2 Commands) SYSTIMESENSITIVE bind option (Db2 Commands)	Additional options: • APPLCOMPAT • APREUSE(WARN) • ARCHIVESENSITIVE(YES|NO) • BUSTIMESENSITIVE(YES|NO) • DBPROTOCOL(DRDACBF) • DESCSTAT(NO|YES) • SYSTIMESENSITIVE(YES|NO)
BIND PACKAGE: COPY bind option (Db2 Commands) DEPLOY bind option (Db2 Commands) LIBRARY bind option (Db2 Commands) MEMBER bind option (Db2 Commands) PACKAGE bind option (Db2 Commands)	The following option values can be delimited. If they are delimited, they can contain mixed-case characters. • *collection-id* • *dbrm-member-name* • *dbrm-library-name* • *package-id*
BIND PLAN and REBIND PLAN: PROGAUTH bind option (Db2 Commands)	Additional options: • PROGAUTH(DISABLE|ENABLE)

Table 8. Changes to existing commands (continued)

Command	Description of enhancements and notes
REBIND TRIGGER PACKAGE: APPLCOMPAT bind option (Db2 Commands) ARCHIVESENSITIVE bind option (Db2 Commands) BUSTIMESENSITIVE bind option (Db2 Commands) SYSTIMESENSITIVE bind option (Db2 Commands)	Additional options: • APPLCOMPAT • ARCHIVESENSITIVE(YES|NO) • BUSTIMESENSITIVE(YES|NO) • SYSTIMESENSITIVE(YES|NO)
-START DB2 (Db2)	Additional options: • LIGHT(CASTOUT)
-START PROFILE	If the PROFILE_AUTOSTART subsystem parameter value is YES, this command is issued automatically when you start Db2.

GUPI

Chapter 10. Utility changes in Db2 11

Use this information to plan for utility changes in Db2 11.

New utilities in Db2 11

Table 9. New utilities

Utility	Description
DSNJU008	The DSNJU008 utility prints the contents of the compression dictionary data set (CDDS).

Utility changes in Db2 11

The following table lists and describes these new and changed options. For information about a new option, see the information for the utility. For information about an option with a changed default value, see Utility release incompatibilities.

Table 10. New and changed utility options

Utility name	Description of enhancements and notes
CHECK INDEX	**New option:** PARALLEL
LOAD	**Changed options:** • Support is added for new IGNORE options: PART, CONV, VALPROC, IDERROR, DUPKEY • Support is added for concurrent access to target tables during LOAD utility processing, with LOAD REPLACE SHRLEVEL REFERENCE. • Support for a wider range of external date formats is added with DATE EXTERNAL (*date-format*) • Support for a wider range of external time formats is added with TIME EXTERNAL (*time-format*) **New options:** • COLGROUP • DRAIN_WAIT • FREQVAL COUNT • 25 • HISTOGRAM • IGNORE • NOCHECKPEND • NUMQUANTILES • NUMCOLS • PARALLEL • RETRY • RETRY_DELAY • SWITCHTIME

© Copyright IBM Corp. 2013, 2019

Table 10. New and changed utility options (continued)

Utility name	Description of enhancements and notes
REBUILD INDEX	**New options:** • HISTOGRAM • NUMQUANTILES • NUMCOLS • PARALLEL
REORG INDEX	**New options:** • HISTOGRAM • NUMQUANTILES • NUMCOLS **Options with changed defaults:** • DRAIN
REORG TABLESPACE	**New options:** • INITCDDS • LISTPARTS • MAPPINGDATABASE • RECLUSTER • COLGROUP • PARALLEL(*num-subtasks*) • SEARCHTIME **Options with changed defaults:** • DRAIN
REPAIR	**New options:** • CATALOG • INSERTVERSIONPAGES • SETCURRENTVERSION
RUNSTATS	**New option:** RESET ACCESSPATH **Changed option:** A default profile is used when USE PROFILE is specified and no profile exists for the target table.
TEMPLATE	**New options:** TIME The TIME option specifies whether UTC or local time is used in expansion of date and time DSN variables. The default value is controlled by the value of the TEMPLATE_TIME subsystem parameter.
UNLOAD	**New option:** PARALLEL

DISPLAY UTILITY output

In Db2 11, the output from the DISPLAY UTILITY command includes the job name and the date and time when the job was submitted.

Related information:

-DISPLAY UTILITY (Db2) (Db2 Commands)
DSNU100I (Db2 Messages)

Utility RBA and LRSN handling

Online utility output
In all migration modes, RBA and LRSN values are displayed in 10-byte format. This 10-byte display is unrelated to migration of the catalog or directory, conversion of individual objects to EXTENDED format, or BSDS conversion. For recovery purposes, this 10-byte format is the preferred input format for Db2. When 10-byte RBA or LRSN values are specified as input to Db2 conversion to 6-byte format is performed internally as needed.

Stand-alone utility output
In all migration modes, formatted RBA and LRSN values are displayed in 10-byte format. The 10-byte formatted display is unrelated to migration of the catalog or directory, conversion of individual objects to EXTENDED format, or BSDS conversion. Dump format data, which might contain RBA or LRSN values, is not consistently shown in 10-byte format. The format of the display represents the actual data at the time of the utility run, and is shown without conversion.

SELECT OUTPUT FROM SYSIBM.SYSLGRNX
A SELECT from SYSIBM.SYSLGRNX displays the LGRSRBA, LGREPBA, LGRSLRSN, and LGRELRSN columns in either 6-byte or 10-byte format. Before CATENFM of SYSLGRNX the data and the display are in 6-byte format. After CATENFM of SYSLGRNX the data and the display are in 10-byte format.

SELECT OUTPUT FROM SYSIBM.SYSCOPY
A SELECT from SYSIBM.SYSCOPY displays the START_RBA and PIT_RBA columns in either 6-byte or 10-byte format. Before CATENFM of SYSCOPY, the data and the display are in 6-byte format but in all migration modes in utility-output, SYSCOPY columns are displayed in 10-byte format. After CATENFM of SYSCOPY the data and the display are in 10-byte format with non-zero digits in low order 3 bytes. Digits in the low order 3 bytes are unrelated to the conversion of the BSDS or conversion of individual objects to EXTENDED format.

Related information:

“Expanded RBA and LRSN log records” on page 19
Chapter 16, “The extended 10-byte RBA and LRSN in Db2 11,” on page 111
What to do before RBA or LRSN limits are reached (Db2 Administration Guide)
SYSLGRNX table (Db2 SQL)
SYSCOPY catalog table (Db2 SQL)

Allocation of image copy data sets

In Db2 11, the COPY utility can use real-time statistics to avoid allocating empty image copy data sets when you specify the CHANGELIMIT or FULL NO options and the appropriate criteria is not met.

If you request that REORG take inline image copies, you can specify a template name for the image copy data sets. In Db2 11, if that template name uses the &PA. or &PART. variable, REORG allocates a separate copy data set for each partition that is being reorganized.

Related information:

Allocation of sequential image copy data sets (Db2 Utilities)

Chapter 11. SQL changes in Db2 11

This version of Db2 for z/OS provides new and changed SQL statements.

New SQL statements in Db2 11

GUPI

Table 11. New SQL statements

SQL statement	Description
CREATE TYPE (array) (Db2 SQL)	The CREATE TYPE (array) SQL statement defines an array type at the current server.
CREATE VARIABLE (Db2 SQL)	The CREATE VARIABLE statement creates a global variable at the current server.
SET CURRENT ACCELERATOR (Db2 SQL)	The SET CURRENT ACCELERATOR changes the value of the CURRENT ACCELERATOR special register.
SET CURRENT APPLICATION COMPATIBILITY (Db2 SQL)	The SET CURRENT APPLICATION COMPATIBILITY statement changes the value of the CURRENT APPLICATION COMPATIBILITY special register.
SET CURRENT TEMPORAL BUSINESS_TIME (Db2 SQL)	The SET CURRENT TEMPORAL BUSINESS_TIME statement changes the value of the CURRENT TEMPORAL BUSINESS_TIME special register.
SET CURRENT TEMPORAL SYSTEM_TIME (Db2 SQL)	The SET CURRENT TEMPORAL SYSTEM_TIME statement changes the value of the CURRENT TEMPORAL SYSTEM_TIME special register.
SET assignment-statement (Db2 SQL)	The SET *assignment-statement* statement is a reclassification of the documentation of the SET *host-variable* and SET *transition-variable* statements into a single statement.

GUPI

SQL statement changes in Db2 11

The following table shows the changes to existing SQL statements. For information about a clause with a changed default value, see Application and SQL release incompatibilities.

GUPI

© Copyright IBM Corp. 2013, 2019

Table 12. Changes to existing SQL statements

SQL statement	Description of enhancements and notes

ALTER FUNCTION (SQL scalar)	**New clauses:**
	`BUSINESS_TIME SENSITIVE`
	`SYSTEM_TIME SENSITIVE`
	`ARCHIVE SENSITIVE`
	`APPLCOMPAT`
	Changed clauses:
	data-type, *data-type2* can include *array-type-name*.

ALTER PROCEDURE (SQL native)	**New clauses:**
	`BUSINESS_TIME SENSITIVE`
	`SYSTEM_TIME SENSITIVE`
	`ARCHIVE SENSITIVE`
	`APPLCOMPAT`
	Changed clauses:
	data-type can include *array-type-name*.

ALTER TABLE	**New clauses:**
	`DROP COLUMN`
	`ENABLE ARCHIVE`
	`DISABLE ARCHIVE`
	Changed clauses:
	ALTER PARTITION clauses that change limit key values now result in pending definition changes. For more information, see Db2 environment, see "ALTER statements that change limit keys are pending changes" in “Application and SQL release incompatibilities” on page 47.

ALTER TABLESPACE	**Changed clauses:** PCTFREE can now include FOR UPDATE *smallint*.
COMMENT	**Changed clauses:** *data-type* can include *array-type-name*.

CREATE FUNCTION (SQL scalar)	**New clauses:**
	`BUSINESS_TIME SENSITIVE`
	`SYSTEM_TIME SENSITIVE`
	`ARCHIVE SENSITIVE`
	`APPLCOMPAT`
	Changed clauses:
	data-type can include *array-type-name*.

CREATE INDEX	**New clauses:**
	`INCLUDE NULL KEYS`
	`EXCLUDE NULL KEYS`

Table 12. Changes to existing SQL statements (continued)

SQL statement	Description of enhancements and notes
CREATE PROCEDURE (external)	**Changed clauses:** *data-type* can include *array-type-name*.

CREATE PROCEDURE (SQL native)	**New clauses:**
	`BUSINESS_TIME SENSITIVE`
	`SYSTEM_TIME SENSITIVE`
	`ARCHIVE SENSITIVE`
	`APPLCOMPAT`
	Changed clauses:
	data-type can include *array-type-name*.

CREATE TABLESPACE	**Changed clauses:** PCTFREE can now include FOR UPDATE *smallint*.

DECLARE GLOBAL TEMPORARY TABLE	**New clauses:**
	`LOGGED`
	`NOT LOGGED`

DROP	**Changed clauses:** *data-type* can include *array-type-name*.
EXECUTE	**Changed clauses:** The object of the USING clause can be an SQL variable, SQL parameter, global variable, or host variable.
FETCH	**Changed clauses:** The object of the INTO clause can be a host variable, an SQL parameter, an SQL variable, a transition variable, or an array element.
GRANT (function or procedure privileges)	**Changed clauses:** *data-type* can include *array-type-name*.
GRANT (type or JAR privileges)	**Changed clauses:** The object of the TYPE clause can be a distinct type or an array type.
OPEN	**Changed clauses:** The object of the USING clause can be an SQL variable, SQL parameter, global variable, or host variable.
REVOKE (function or procedure privileges)	**Changed clauses:** *data-type* can include *array-type-name*.
REVOKE (type or JAR privileges)	**Changed clauses:** The object of the TYPE clause can be a distinct type or an array type.

Table 12. Changes to existing SQL statements (continued)

SQL statement	Description of enhancements and notes
SELECT INTO	**Changed clauses:** The object of the INTO clause can be a host variable, a global variable, an SQL parameter, an SQL variable, a transition variable, or an array element.
SET PATH	**Changed clauses:** The SYSTEM PATH now includes the schemas "SYSIBM", "SYSFUN", "SYSPROC", "SYSIBMADM".
SQL statement with subselect	**Changed clauses:** *collection-derived-table* is added to *table-reference* in the FROM clause of a *subselect*. **Other changes:** A user-defined function that is defined with MODIFIES SQL DATA can be invoked in a subselect.
VALUES INTO	**Changed clauses:** The object of the INTO clause can be a host variable, a global variable, an SQL parameter, an SQL variable, a transition variable, or an array element.

GUPI

New built-in functions in Db2 11

Db2 11 introduces new built-in functions that improve the power of the SQL language. The following table shows the new built-in functions.

GUPI

Table 13. New functions

Function name	Description
ARRAY_AGG (Db2 SQL)	The ARRAY_AGG function returns an array in which each value of the input set is assigned to an element of the array.
ARRAY_DELETE (Db2 SQL)	The ARRAY_DELETE function deletes elements from an array.
ARRAY_FIRST (Db2 SQL)	The ARRAY_FIRST function returns the minimum array index value of an array.
ARRAY_LAST (Db2 SQL)	The ARRAY_LAST function returns the maximum array index value of an array.
ARRAY_NEXT (Db2 SQL)	The ARRAY_NEXT function returns the next larger array index value, relative to a specified array index value.
ARRAY_PRIOR (Db2 SQL)	The ARRAY_PRIOR function returns the next smaller array index value, relative to a specified array index value.
BLOCKING_THREADS (Db2 SQL)	The BLOCKING_THREADS function returns a table that contains one row for each lock or claim that threads hold against specified databases.

Table 13. New functions (continued)

Function name	Description
CARDINALITY (Db2 SQL)	The CARDINALITY function returns the number of elements in an array.
CHAR9 (Db2 SQL)	The CHAR9 function returns a fixed-length character string representation of the argument. The CHAR9 function is intended for compatibility with previous releases of Db2 for z/OS that depend on the result format that is returned for decimal input values in Version 9 and earlier. **Important:** For portable applications that might run on platforms other than Db2 for z/OS, use the CHAR function instead. Other Db2 family products do not support the CHAR9 function.
MAX_CARDINALITY (Db2 SQL)	The MAX_CARDINALITY function returns the maximum number of elements that an array can contain.
MEDIAN	The MEDIAN function returns the median of a set of numbers. This function can run only on an accelerator server.
TRIM_ARRAY (Db2 SQL)	The TRIM_ARRAY function deletes elements from the end of an ordinary array.
VARCHAR9 (Db2 SQL)	The VARCHAR9 function returns a fixed-length character string representation of the argument. The VARCHAR9 function is intended for compatibility with previous releases of Db2 for z/OS that depend on the result format that is returned for decimal input values in Version 9 and earlier. **Important:** For portable applications that might run on platforms other than Db2 for z/OS, use the VARCHAR function instead. Other Db2 family products do not support the VARCHAR9 function.

GUPI

New reserved words in Db2 11

Db2 11 introduces the following SQL reserved words:

ARRAY
ARRAY_EXISTS
VERSIONING

For the complete list, see Reserved words (Db2 SQL).

Chapter 12. Catalog changes in Db2 11

This version of Db2 for z/OS includes changed catalog tables, new catalog tables, and new indexes on catalog tables.

New catalog tables in Db2 11

GUPI

Table 14. New catalog tables

Catalog table name	Description
SYSINDEXCLEANUP catalog table (Db2 SQL)	The rows in the SYSIBM.SYSINDEXCLEANUP table specify time windows to control index cleanup processing. Each row specifies a time window to enable or disable the cleanup of pseudo-deleted index entries for specific database objects.
SYSQUERYPREDICATE catalog table (Db2 SQL)	The SYSIBM.SYSQUERYPREDICATE table contains information about predicates for queries in the SYSIBM.SYSQUERY table that have been identified for extended optimization. It correlates to the SYSIBM.SYSQUERY table by the QUERYID column.
SYSQUERYSEL catalog table (Db2 SQL)	The SYSIBM.SYSQUERYSEL table contains information about the selectivity of predicates for queries in the SYSIBM.SYSQUERY table that have been identified for extended optimization. It correlates to the SYSIBM.SYSQUERY table by the QUERYID column.
SYSSTATFEEDBACK catalog table (Db2 SQL)	The SYSIBM.SYSTATFEEDBACK table contains information about missing or conflicting catalog statistics for SQL statements.
SYSVARIABLEAUTH catalog table (Db2 SQL)	The SYSIBM.SYSVARIABLEAUTH table contains one row for each privilege of each authorization ID that has privileges on a global variable.
SYSVARIABLES_DESC catalog table (Db2 SQL)	The SYSIBM.SYSVARIABLES_DESC table is an auxiliary table for the SYSIBM.SYSVARIABLES table.
SYSVARIABLES catalog table (Db2 SQL)	The SYSIBM.SYSVARIABLES table contains one row for each global variable that is created.
SYSVARIABLES_TEXT catalog table (Db2 SQL)	The SYSIBM.SYSVARIABLES_TEXT table is an auxiliary table for the DEFAULTTEXT column of the SYSIBM.SYSVARIABLES table.

GUPI

Changed catalog tables in Db2 11

The following table shows a list of the new columns and the existing columns that are revised. Revisions to columns include new column descriptions, new values for a column, changed data types, changed column lengths, or both changed data types and lengths.

GUPI

Table 15. Summary of new and revised catalog table columns

Catalog table name	New column	Revised column
SYSCHECKS catalog table (Db2 SQL)		RBA

© Copyright IBM Corp. 2013, 2019

Table 15. Summary of new and revised catalog table columns (continued)

Catalog table name	New column	Revised column
SYSCOPY catalog table (Db2 SQL)	MODECREATED	• PIT_RBA • START_RBA • ICTYPE • STYPE • TTYPE
SYSDATATYPES catalog table (Db2 SQL)	• ARRAYLENGTH • ARRAYINDEXTYPEID • ARRAYINDEXTYPELEN • ARRAYINDEXSUBTYPE	• LENGTH • METATYPE • SCALE • SUBTYPE
SYSDEPENDENCIES catalog table (Db2 SQL)	DVERSION	• BTYPE • DTYPE
SYSINDEXES catalog table (Db2 SQL)		• CLUSTERED • CLUSTERRATIO • COPYLRSN • SPARSE
SYSINDEXES_HIST catalog table (Db2 SQL)		• AVGKEYLEN • CLUSTERRATIOF • DATAREPEATFACTORF
SYSINDEXPART catalog table (Db2 SQL)	RBA_FORMAT	• AVGKEYLEN • CARDF • FAROFFPOSF • NEAROFFPOSF
SYSINDEXPART_HIST catalog table (Db2 SQL)		• AVGKEYLEN • CARDF • FAROFFPOSF • NEAROFFPOSF
SYSINDEXSPACESTATS catalog table (Db2 SQL)		COPYUDPATELRSN
SYSINDEXSTATS catalog table (Db2 SQL)		• CLUSTERRATIO • CLUSTERRATIOF • DATAREPEATFACTORF • KEYCOUNT • KEYCOUNTF
SYSINDEXSTATS_HIST catalog table (Db2 SQL)		• CLUSTERRATIOF • DATAREPEATFACTORF • KEYCOUNTF

Table 15. Summary of new and revised catalog table columns (continued)

Catalog table name	New column	Revised column
SYSPACKAGE catalog table (Db2 SQL)	• APPLCOMPAT • ARCHIVESENSITIVE • BUSTIMESENSITIVE • DESCSTAT • EXTSEQNO	• APREUSE • DBPROTOCOL • SYSTIMESENSITIVE
SYSPACKCOPY catalog table (Db2 SQL)	• APPLCOMPAT • ARCHIVESENSITIVE • BUSTIMESENSITIVE • DESCSTAT • EXTSEQNO • SYSTIMESENSITIVE	• DBPROTOCOL • RECORDTEMPORALHIST • SYSTIMESENSITIVE
SYSPACKDEP catalog table (Db2 SQL)		BTYPE
SYSPACKSTMT catalog table (Db2 SQL)	EXPANSION_REASON	
SYSPARMS catalog table (Db2 SQL)		• CCSID • ENCODING_SCHEME • LENGTH • SUBTYPE
SYSPENDINGDDL catalog table (Db2 SQL)	• COLNAME • COLUMN_KEYWORD • PARTITION • PARTITION_KEYWORD	• CREATEDTS • OBJTYPE • OPTION_KEYWORD • OPTION_VALUE • STATEMENT_TYPE
SYSPLANDEP catalog table (Db2 SQL)		BTYPE
SYSPLAN catalog table (Db2 SQL)	PROGAUTH	
SYSQUERY catalog table (Db2 SQL)	• ACCESS_PATH_HINT • OPTION_OVERRIDE • SELECTIVITY_VALID • SELECTVTY_OVERRIDE	
SYSQUERYPLAN catalog table (Db2 SQL)	EXPANSION_REASON	
SYSRESAUTH catalog table (Db2 SQL)		QUALIFIER
SYSROUTINES catalog table (Db2 SQL)		COMMIT_ON_RETURN
SYSSEQUENCES catalog table (Db2 SQL)	• SEQNAME • SEQSCHEMA	

Table 15. Summary of new and revised catalog table columns (continued)

Catalog table name	New column	Revised column
SYSTABLEPART catalog table (Db2 SQL)	• PCTFREE_UDP • PCTRFREE_UPD_CALC • RBA_FORMAT	• LIMITKEY
SYSTABLES catalog table (Db2 SQL)	• ARCHIVING_SCHEMA • ARCHIVING_TABLE • STATS_FEEDBACK	• RBA1 • RBA2 • VERSION
SYSTABLESPACESTATS catalog table (Db2 SQL)	• UPDATESIZE • LASTDATACHANGE	COPYUPDATELRSN

GUPI

New and changed indexes in Db2 11

Db2 11 introduces the following new indexes.

Table 16. New indexes in Db2 11

Catalog table	New index
SYSCHECKDEP	DSNSDX01
SYSCHECKS	DSNSCX01
SYSCHECKS2	DSNCHX01
SYSCOPY	DSNUCH01
SYSINDEXCLEANUP	DSNICX01
SYSINDEXSPACESTATS	DSNRTX02
SYSOBD_AUX	DSNOB03
SYSQUERYPREDICATE	• DSNQEX01 • DSNQEX02
SYSQUERYSEL	• DSNQLX01 • DSNQLX02
SYSSEQUENCES	DSNSQX03
SYSSTATFEEDBACK	• DSNSFX01 • DSNSFX02 • DSNSFX03
SYSSTRINGS	DSNSSX01
SYSTABLESPACESTATS	DSNRTX01
SYSTABSTATS	DSNTTX02

Table 16. New indexes in Db2 11 (continued)

Catalog table	New index
SYSVARIABLEAUTH	• DSNVAX01 • DSNVAX02 • DSNVAX03
SYSVARIABLES	DSNOVX01
SYSVARIABLES_DESC	DSNOVX03
SYSVARIABLES_TEXT	DSNOVX02

Db2 11 introduces changes to the following indexes:

Table 17. Changed indexes in Db2 11

Catalog table	Changed index
SYSXMLTYPMOD	DSNTMX01
SYSXMLTYPMSCHEMA	DSNMSX01

For more information about catalog index formats in Db2 11, see Catalog indexes (Db2 SQL).

New built-in global variables

Db2 11 introduces the following built-in global variables:

- CLIENT_IPADDR
- GET_ARCHIVE
- MOVE_TO_ARCHIVE

For descriptions, see Built-in global variables (Db2 SQL).

Chapter 13. EXPLAIN table changes in Db2 11

Before you can use EXPLAIN, you must create an EXPLAIN table to hold the captured information. Certain optimization tools also create and use their own instances of the EXPLAIN tables.

PSPI

For the current formats, and sample CREATE TABLE statements for PLAN_TABLE and the other EXPLAIN tables, see member DSNTESC of the *prefix*.SDSNSAMP library. For the complete set of column descriptions for each table, see EXPLAIN tables (Db2 Performance) and Input tables (Db2 Performance).

Important: It is best to convert EXPLAIN tables to Db2 11 format during migration, or soon after migration. In Db2 11, the EXPLAIN function supports tables in Db2 11 or Db2 10 formats only. However, Db2 10 format EXPLAIN tables are deprecated. If you invoke EXPLAIN and Db2 10 tables are used, Db2 issues SQL code +20520. If tables of an unsupported format are found, Db2 issues SQL code -20008 and the EXPLAIN operation fails. You can call the ADMIN_EXPLAIN_MAINT stored procedure to create EXPLAIN tables, upgrade them to the format for the current Db2 release, or complete other maintenance tasks. See ADMIN_EXPLAIN_MAINT stored procedure (Db2 SQL) for information about using the *action* input parameter to request each of these tasks.

DSNTIJG creates no EXPLAIN tables with SYSIBM qualifier

The DSNTIJSG installation job no long contains CREATE statements for EXPLAIN tables with the SYSIBM qualifier. Db2 no longer uses these tables.

Universal table spaces for EXPLAIN tables

In Db2 11, the table space type for the EXPLAIN tables that are created by the supplied sample jobs and the ADMIN_EXPLAIN_MAINT stored procedure are created as partition-by-growth table spaces, instead of segmented (non-UTS) table spaces. For more information about table space types, see Table space types and characteristics in Db2 for z/OS (Db2 Administration Guide).

New and changed EXPLAIN tables

The following table lists only new and changed columns for the following EXPLAIN tables:

Table 18. List of new and changed EXPLAIN tables

EXPLAIN table	Changes
PLAN_TABLE (Db2 Performance)	New columns: • SCAN_DIRECTION • EXPANSION_REASON Changed columns: • MERGN (new values) • PRIMARY_ACCESSTYPE
DSN_PREDICAT_TABLE (Db2 Performance)	New columns: • EXPANSION_REASON Changed columns: • ADDED_PRED (new values)
DSN_PREDICATE_SELECTIVITY table (Db2 Performance)	New table.

© Copyright IBM Corp. 2013, 2019

Table 18. List of new and changed EXPLAIN tables (continued)

EXPLAIN table	Changes
DSN_PTASK_TABLE (Db2 Performance)	New columns: • EXPANSION_REASON
DSN_QUERYINFO_TABLE (Db2 Performance)	New columns: • EXPANSION_REASON
DSN_QUERY_TABLE (Db2 Performance)	New columns: • EXPANSION_REASON
DSN_SORTKEY_TABLE (Db2 Performance)	New columns: • EXPANSION_REASON
DSN_SORT_TABLE (Db2 Performance)	New columns: • EXPANSION_REASON
DSN_STATEMNT_TABLE (Db2 Performance)	New columns: • EXPANSION_REASON
DSN_STRUCT_TABLE (Db2 Performance)	New columns: • EXPANSION_REASON
DSN_VIEWREF_TABLE (Db2 Performance)	New columns: • EXPANSION_REASON
DSN_VIRTUAL_INDEXES (Db2 Performance)	New columns: • KEYTARGET_COUNT • UNIQUE_COUNT • IX_EXTENSION_TYPE • DATAREPEATFACTORF • SPARSE
DSN_VIRTUAL_KEYTARGETS (Db2 Performance)	New table.

PSPI

Related concepts
Investigating SQL performance by using EXPLAIN (Db2 Performance)
Input tables (Db2 Performance)
Related tasks
Migration step 24: Convert EXPLAIN tables to the current format (Db2 Installation and Migration)
Related reference
EXPLAIN (Db2 SQL)
EXPLAIN tables (Db2 Performance)

Chapter 14. IFCID changes in Db2 11

Db2 11 introduces IFCID changes.

This information briefly describes the new IFCIDs and the changes to the existing IFCIDs for each new function. For a detailed description of the fields in each IFCID record, refer to the mapping macros data set library DSN1110.SDSNMACS.

New trace records

PSPI

The following tables gives an overview of new IFCIDs. Serviceability trace records are not included.

Table 19. New IFCIDs

IFCID	Trace	Class	Mapping macro	Description
0377	None	None	DSNDQW05	Records information about indexes for which pseudo-deleted index entries are automatically cleaned up.
0382	Accounting, Monitor	3, 8	DSNDQW05	Records suspend operations for parallel task synchronization.
0383	Accounting, Monitor	3, 8	DSNDQW05	Records resume operations after parallel task synchronization.

PSPI

Changes to selected trace records

PSPI

The following table gives an overview of changes to specific IFCIDs. Changes to IFCID 0106, the system parameters record, are not included.

Table 20. Changed IFCIDs

Enhancement	IFCID	Description of changes
Array support	0002, 0225	Fields are added to record information about storage use by arrays.
Autonomous transactions	0003, 0239	Fields are added to record information about autonomous transactions, and on their effect on parallel groups.
Application compatibility	0366	Field values are added to record incompatibilities between Db2 11 and previous versions.
Castout enhancements	0230, 0256	Fields are added to record class castout queue threshold values, which are based on the number of pages.
Hybrid Transactional Analytical Processing support for accelerated queries (APAR PI90239)	0002	Fields are added to monitor the effect of Hybrid Transactional Analytical Processing.

© Copyright IBM Corp. 2013, 2019

Table 20. Changed IFCIDs (continued)

Enhancement	IFCID	Description of changes
IBM zHyperLink exploitation for database synchronous read I/O (APAR PH01123)	0002	Fields are added to record: • The number of successful synchronous read I/O operations that were performed using IBM zHyperLink processing. • The number of synchronous read I/O operations that resulted in disk cache hits, and did not use IBM zHyperLink processing.
	0003, 0148	Fields are added to record: • Wait time for synchronous read I/O operations that resulted in disk cache hits • The number of synchronous read I/O operations that resulted in disk cache hits.
	0003, 0148, 0239	Fields are added to record: • The number of synchronous read I/O operations that used IBM zHyperlink processing. • The number of synchronous read I/O operations that resulted in disk cache hits. • The amount of CPU time for successful synchronous read I/O operations that used IBM zHyperlink processing.
	0006	A flag is added to indicate whether a synchronous read I/O request occurred with or without IBM zHyperLink processing.
	0007	A flag is added to indicate whether a synchronous read I/O request with IBM zHyperLink processing was successful.
	0199	Fields are added to record the average and maximum delay and total number of pages read with synchronous read I/O and IBM zHyperLink processing.
Larger RBA and LRSN support	0001, 0032, 0034, 0043, 0114, 0119, 0143, 0144, 0185, 0188, 0203, 0204, 0206, 0207, 0208, 0209, 0235, 0261, 0306, 0313, 0335	Fields that contain RBAs and LRSNs are expanded from 6 or 8 bytes to 10 bytes. Fields other than QW0204UR are moved to avoid changes in other offsets. Field QW0204UR is expanded but not moved. Therefore, the offsets of all fields in IFCID 0204 that follow it are changed.
Parallelism performance enhancements	0002, 0003, 0316, 0401	Fields are added to track the effect of changes to the degree of parallelism after parallel system negotiation that occurs because of resource constraints.
Statement-level monitoring for accelerated Hybrid Transactional and Analytical Processing (HTAP) support (APAR PH00574)	0003, 0316	Fields are added to record: • The amount of time in milliseconds that statements waited for execution on an accelerator because the statements needed changes that were not yet applied to a referenced table. • The number of times that statement execution did not complete because changes were not applied to a referenced table before the delay protocol time period expired.

Table 20. Changed IFCIDs (continued)

Enhancement	IFCID	Description of changes
Temporal support	0053, 0058, 0060, 0061, 0065, 0316, 0401	Fields are added to indicate the impacts of the CURRENT TEMPORAL BUSINESS_TIME special register, the CURRENT TEMPORAL SYSTEM_TIME special register, and the SYSIBMADM.GET_ARCHIVE built-in global variable.
Miscellaneous changes	0002	Fields are added to record: • The amount of storage that is allocated to shareable, static SQL statements. • The number of pages that are written to disk through group-buffer-pool write-around protocol
	0002, 0003	Fields are added to record: • Additional IRLM system information
	0003	• Fields are added to record the wait time and number of waits for parallel queries to synchronize. • Existing fields that record accumulated wait time for write I/O that is done under another thread and the number of wait trace events that were processed for write I/O under another thread now include time that is consumed by DFSMS Media Manager calls when buffers are written to disk.
	0003, 0148, 0239	Sysplex query parallelism is no longer supported. Fields that are related to Sysplex query parallelism are not populated.
	0127, 0128	Fields are added to record the time spent waiting for buffer manager force write I/O.
	0148	A field is added to record an additional DDF enclave.
	0201	Fields are added to record the virtual pool size, minimum pool size, maximum pool size, and frame size before and after the ALTER BUFFERPOOL command.
	0217, 0225	Fields are added to record IRLM private storage limits.
	0225	The following changes are made: • Fields are added to record the common storage that is used by log manager buffers and control structures. • Data is no longer recorded for the number of log manager write buffer frames in auxiliary storage. • Fields that record shareable storage use for SQL statements are moved and expanded to 8 bytes. The corresponding 4-byte fields are no longer used.
	0239	• Package detail information is rolled up and associated with a rollup QPAC record if an accounting class 10 trace or a monitor class 10 trace is enabled. • Fields are added to record the wait time and number of waits for parallel queries to synchronize.

Table 20. Changed IFCIDs (continued)

Enhancement	IFCID	Description of changes
	0313	New fields are added to record: • User IDs that are longer than 16 bytes • Transaction names that are longer than 32 bytes • Workstation names that are longer than 18 bytes

PSPI

Chapter 15. New and changed subsystem parameters in Db2 11

This version of Db2 for z/OS includes new and changed subsystem parameters.

New subsystem parameters in Db2 11

GUPI

Table 21. New subsystem parameters

Subsystem Parameter	Description
APPL COMPAT LEVEL field (APPLCOMPAT subsystem parameter) (Db2 Installation and Migration)	Determines the default value for application compatibility.
AUTH EXIT CACHE REFR (AUTHEXIT_CACHEREFRESH subsystem parameter) (Db2 Installation and Migration)	Specifies whether the package authorization cache, routine authorization cache, and dynamic statement cache entries are to be refreshed, when access control authorization exit is active and the user profile is changed in RACF.
AUTH EXIT CHECK (AUTHEXIT_CHECK subsystem parameter) (Db2 Installation and Migration)	Specifies whether the owner or the primary authorization ID is to be used for authorization checks, when the access control authorization exit is active.
CDDS_MODE in macro DSN6LOGP (Db2 Installation and Migration)	Specifies whether a member is part of a Db2 data sharing group that participates in a GDPS Continuous Availability with zero data loss environment.
CDDS_PREFIX in macro DSN6LOGP (Db2 Installation and Migration)	Specifies the prefix for the name of the data set that contains dictionaries that Db2 uses to decompress compressed log records when a proxy data sharing group in an GDPS Continuous Availability with zero data loss environment captures the log records on behalf of a source data sharing group.
DISALLOW_SEL_INTO_UNION in macro DSN6SPRM (Db2 Installation and Migration)	Specifies whether to disallow a UNION or UNION ALL as the outermost from-clause of a SELECT INTO statement, which is invalid SQL syntax. Such statements were inadvertently allowed before Db2 12. However, starting in Db2 12 they are disallowed by default.
FLASHCOPY XRCP field (FLASHCOPY_XRCP subsystem parameter) (Db2 Installation and Migration)	If z/OS DFSMSdss support for RPFC in XRC is installed and enabled, specifies whether a data set level FlashCopy operation is allowed to a primary volume in an XRC relationship.
INDEX CLEANUP THREADS field (INDEX_CLEANUP_THREADS subsystem parameter) (Db2 Installation and Migration)	Specifies the maximum number of threads that can be created to process the cleanup of pseudo-deleted index entries on this subsystem or data sharing member. Pseudo-deleted entries in an index are those that are logically deleted but still physically present in the index. This parameter works in conjunction with the SYSIBM.SYSINDEXCLEANUP catalog table, which controls cleanup processing of pseudo-deleted index entries.

© Copyright IBM Corp. 2013, 2019

Table 21. New subsystem parameters (continued)

Subsystem Parameter	Description
LIKE BLANK INSIGNIFICANT field (LIKE_BLANK_INSIGNIFICANT subsystem parameter) (Db2 Installation and Migration)	Specifies whether the LIKE predicate should ignore trailing blanks within fixed length character strings.
MAX IN-MEMORY SORT SIZE field (MAXSORT_IN_MEMORY subsystem parameter) (Db2 Installation and Migration)	Specifies the maximum size in kilobytes of storage to allocate for sorting the results of each query that specifies the ORDER BY clause, the GROUP BY clause, or both. The storage is allocated only during the query processing. Increasing the value in this field can improve performance of such queries.
OBJECT CREATE FORMAT field (OBJECT_CREATE_FORMAT subsystem parameter) (Db2 Installation and Migration)	Specifies whether Db2 is to create new table spaces and indexes to use a basic or extended log record format.
MAX DEGREE FOR DPSI (PARAMDEG_DPSI subsystem parameter) (Db2 Installation and Migration)	Specifies the maximum degree of parallelism that is allowed for a parallel group where a data partitioned secondary index (DPSI) is used to drive parallelism.
MAX UTILS PARALLELISM field (PARAMDEG_UTIL subsystem parameter) (Db2 Installation and Migration)	Specifies the maximum degree of parallelism that is allowed when a Db2 utility is driving parallelism.
PERCENT FREE FOR UPDATE field (PCTFREE_UPD subsystem parameter) (Db2 Installation and Migration)	Specifies the default percentage of each page that Db2 will leave as free space for updates when you create a table space without specifying the FOR UPDATE clause of the PCTFREE option, and a table within that table space is populated. Reserved free space is consumed only by update operations. It is reserved when the table loaded or reorganized or populated by means of SQL INSERT statements. It is not reserved for a LOB table space, a hash table space, a table space that is implicitly created for an XML column, or a table space in a work file database.
PACKAGE RELEASE COMMIT field (PKGREL_COMMIT subsystem parameter) (Db2 Installation and Migration)	Specifies whether the following operations on a package that is bound as RELEASE(DEALLOCATE) are permitted while the package is active and allocated by Db2: • BIND and REBIND requests, including automatic rebinds • DDL changes to objects that are statically referenced by the package
PREVENT ALTER LIMITKEY field (PREVENT_ALTERTB_LIMITKEY subsystem parameter) (Db2 Installation and Migration)	Determines whether Db2 disallows altering the limit key by using an ALTER TABLE statement for index-controlled partitioned table spaces. This alter operation places the table space in REORG-pending (REORP) restrictive status, and the data is unavailable until the affected partitions are reorganized. Use PREVENT_ALTERTB_LIMITKEY to avoid this data unavailability.
PROFILE AUTOSTART field (PROFILE_AUTOSTART subsystem parameter) (Db2 Installation and Migration)	Specifies whether START PROFILE command processing is automatically initiated as part of Db2 start processing.

Table 21. New subsystem parameters (continued)

Subsystem Parameter	Description
REMOTE_COPY_SW_ACCEL	Specifies whether Db2 uses software to control the remote copy process for active log output in peer-to-peer remote copy (PPRC) environments. The default value is DISABLE. This parameter is also added to DB2 10.
REORG DROP PBG PARTS field (REORG_DROP_PBG_PARTS subsystem parameter) (Db2 Installation and Migration)	Specifies whether REORG is to remove trailing empty partitions when operating on an entire partitioned by growth (PBG) table space. An empty trailing partition occurs when REORG moves all data records from a partition into lower numbered partitions. This parameter is meaningful only when REORG is run against an entire PBG table space. It is ignored for REORGs of non-PBG table spaces and for partition-level REORGs of PBG table spaces.
REORG MAPPING DB field (REORG_MAPPING_DATABASE subsystem parameter) (Db2 Installation and Migration)	Specifies the default database in which REORG TABLESPACE SHRLEVEL CHANGE should implicitly create the mapping table.
STATISTICS COLGROUP DATA SORT STG LIMIT field (STATCLGSRT subsystem parameter) (Db2 Installation and Migration)	Specifies the maximum amount of storage that Db2 can use to avoid a sort for a RUNSTATS utility or inline statistics job when collecting frequency statistics on one or more single-column column groups, defined with the COLGROUP option.
STATISTICS FEEDBACK field (STATFDBK_SCOPE subsystem parameter) (Db2 Installation and Migration)	Specifies the scope of SQL statements for which Db2 is to recommend statistics.
TEMPLATE TIME field (TEMPLATE_TIME subsystem parameter) (Db2 Installation and Migration)	Specifies the default setting for the TIME option of the Db2 utilities TEMPLATE statement, that is the value to use when the TEMPLATE statement does not specify the TIME option.
UTILITY OBJECT CONVERSION field (UTILITY_OBJECT_CONVERSION subsystem parameter) (Db2 Installation and Migration)	Specifies whether Db2 utilities that accept the RBALRSN_CONVERSION option will convert existing table spaces and indexes to 6-byte page format, to a 10-byte page format, or perform no conversion.
AGENT LEVEL THRESHOLD field (WFSTGUSE_AGENT_THRESHOLD subsystem parameter) (Db2 Installation and Migration)	Specifies the percentage of the space available in the work file database on this Db2 subsystem or data sharing member that can be consumed by a single agent before a warning message is issued. Space in the work file database can be configured for temporary (DGTT-oriented) work and for sort work, including CGTTs and trigger transition tables.
SYSTEM LEVEL THRESHOLD field (WFSTGUSE_SYSTEM_THRESHOLD subsystem parameter) (Db2 Installation and Migration)	Specifies the percentage of the space available in the work file database on this Db2 subsystem or data sharing member that can be consumed by all agents before a warning message is issued. Space in the work file database can be configured for temporary (DGTT-oriented) work and for sort work, including CGTTs and trigger transition tables.
DB2 zHyperLinks SCOPE field (ZHYPERLINK subsystem parameter) (Db2 Installation and Migration)	Specifies the default scope of the zHyperLink protocol for I/O requests.

Changed subsystem parameters in Db2 11

GUPI

The following table shows the existing subsystem parameters that have new and changed options. For information about an option with a changed default value, see Other release incompatibilities.

GUPI

Table 22. Changes to existing subsystem parameters

Subsystem parameter	Description of enhancements and notes
AUTO BIND field (ABIND subsystem parameter) (Db2 Installation and Migration)	COEXIST behavior is used even when YES is specified. That is, no automatic bind occurs in release coexistence or fallback scenarios when a plan or package that was automatically bound in Db2 10 runs again in Db2 11. This change prevents repeating automatic remigration binds each time that a plan or package runs again on a different release during coexistence or fallback scenarios. For more information, see Automatic binds in coexistence (Db2 Installation and Migration).
DSMAX field (DSMAX subsystem parameter) (Db2 Installation and Migration)	The upper limit for the maximum number of data sets that can be open at one time is changed from 100,000 to 200,000.
EDM DBD CACHE field (EDMDBDC subsystem parameter) (Db2 Installation and Migration)	The upper limit for the minimum size (in KB) of the DBD cache that is to be used by EDM is changed from 2,097,152 to 4,194,304.
EDM SKELETON POOL SIZE field (EDM_SKELETON_POOL subsystem parameter) (Db2 Installation and Migration)	The upper limit of the minimum size of the EDM skeleton pool (in KB) is changed from 2,097,152 to 4,194,304.
EDM STATEMENT CACHE field (EDMSTMTC subsystem parameter) (Db2 Installation and Migration)	The upper limit for the size (in KB) of the statement cache that is to be used by the EDM is changed from 1,048,576 to 4,194,304.
MAX KEPT DYN STMTS field (MAXKEEPD subsystem parameter) (Db2 Installation and Migration)	The upper limit for the total number of prepared, dynamic SQL statements that can be saved past a commit point by all threads in the system using the KEEPDYNAMIC(YES) bind option is changed is changed from 65,535 to 204,800.
REORG PART SORT NPSI field (REORG_PART_SORT_NPSI subsystem parameter) (Db2 Installation and Migration)	The default value is changed from NO to AUTO.
SUBQ_MIDX	The default value is changed from DISABLE to ENABLE.

Removed subsystem parameters in Db2 11

The following tables lists subsystem parameters that are removed from this version of Db2 for z/OS. Refer to the information for the earlier version for detailed descriptions of the removed subsystem parameters.

Table 23. Removed subsystem parameters

Subsystem parameter	Setting used in Db2 11 behavior
ASSIST	NO
COORDNTR	NO
DISABSCL	NO
DISALLOW_DEFAULT_COLLID	NO
MSVGP and MSVGP2	These devices are not recognized in Db2 11.
OJPERFEH	YES
OPTIOPIN	ENABLE
OTPIOWGT	ENABLE
OPTIXIO	ON
PGRNGSCR	YES
PTCDIO	OFF
RETVLCFK	NO
SEQCACH	SEQ
SEQPRES	YES
SMSDCFL	blank
SMSDCIX	blank
STATCLUS	ENHANCED

GUPI

Chapter 16. The extended 10-byte RBA and LRSN in Db2 11

Beginning in Db2 11, Db2 uses 10-byte RBA and LRSN values.

Differences between the 6-byte and 10-byte formats

The terms "basic" and "extended" are sometimes used to refer to the 6-byte and 10-byte formats. When these terms are used, *basic format* refers to the 6-byte format, and *extended format* refers to the extended 10-byte format.

Conversion of RBA values

A 6-byte RBA value is converted to the 10-byte format value by adding zeros to the 4 most significant bytes. That is, the zeros are added to the left side of the value, as shown in the following table.

6-byte RBA value:	10-byte RBA value:
112233445566	**00000000**112233445566

Conversion LRSN values

A 6-byte LRSN value is converted to a 10-byte value by adding one zero byte to the left side and 3 bytes added to the right side of the value, as shown in the following table.

6-byte LRSN value:	10-byte LRSN value:
112233445566	**00**112233445566**000000**

The 3 bytes on the right side might be might be zero or x'FF', depending on the situation. For the beginning of an LRSN range, zeros are used. For the end of an LRSN range, x'FF' is used.

Internally, the values that are kept in memory are all 10 bytes, except when they need to be externalized to structures that remain in the 6-byte format. The values are stored internally as 10 bytes even in conversion mode. The conversion from the 10-byte values to 6-byte format is done at end points, such as when a log record is written, or when the PGLOGRBA field in a data or index page is updated.

When 10-byte format values are externalized

In Db2 11, the extended RBA and LRSN values might be externalized before objects are converted in certain contexts:

Messages

In Db2 11, all messages use 10-byte RBA and LRSN values in all migration modes, so that all messages have consistent formats. Sometimes Db2 needs an LRSN value that is not associated with a specific update. In this case, a log record with a matching LRSN might not exist. Such LRSN values are often generated with non-zero precision in the last 3 bytes, regardless of mode. Such full-precision 10-byte values might be seen in message output.

Database objects

The RBA (non-data sharing) or LRSN (data sharing) of the last change is stored in each page of every table and index.

When objects are in the basic format, the stored RBA or LRSN values are always 6 bytes. In the extended format, the stored RBA or LRSN values are 10 bytes. An installation typically converts objects from basic to extended format by using the REORG utility, but other methods exist. In addition, after migration to NFM the installation can decide which format is used to create new database objects. Database objects can be converted from extended to basic unless prohibited by a subsystem parameter.

© Copyright IBM Corp. 2013, 2019

Objects in basic format cannot be updated when the RBA or LRSN value is beyond the 6-byte range. For data sharing groups, the 6-byte LRSN range applies to the entire group and the 6-byte RBA range applies to each member. The 6-byte LRSN range does not apply to non-data sharing environments.

Recovery logs

The log records are assigned RBA values so that they can be located. In a data sharing environment, each log record has an associated LRSN value that is based on the time the log record was created. The LRSN value can be used to sequence log records from multiple members in a data sharing group.

In conversion mode, the log content and format are identical to the DB2 10 format. All values that are passed to other Db2 components internally are 10-byte values padded with zeros. To all components outside of the log manager, the log always appears to be in the 10-byte format. Conversion of the log content to the new format that supports 10-byte RBA and LRSN values is completed when the installation converts the BSDSs to the 10-byte format. These two actions must be completed in lock step because the old BSDS format cannot accommodate larger RBA and LRSN values. For more information, see How RBA and LRSN values are displayed (Db2 Administration Guide) and SYSLGRNX table (Db2 SQL).

Bootstrap data sets (BSDS)

The BSDS contains the LRSN and RBA values that bound each active and archive log data and a number of others that have various purposes. Db2 11 supports a data sharing group with members using different BSDS formats indefinitely. However, after a BSDS is converted, it cannot be restored to 6-byte format because it might contain information that cannot be truncated to the 6-byte format.



Attention: In Db2 subsystems that are not data sharing members, if Db2 is already at risk of reaching the 6-byte RBA limit, it is strongly recommended that you first convert all catalog and directory objects, then convert all user objects to the 10-byte RBA format, before you convert the BSDS.

Before you can convert the BSDS, the Db2 subsystem or data sharing member must be started in Db2 11 new-function mode. For instructions, see Migrating from enabling-new-function mode to new-function mode (Db2 Installation and Migration).

In data sharing, you must convert the BSDS for each Db2 member separately.

At migration to Db2 12, you cannot start Db2 12 until the BSDS is converted to use the 10-byte RBA and LRSN formats. You can convert the BSDS before or during the Db2 12 migration process.



Attention: After the BSDS is converted to the 10-byte format, Db2 stops issuing messages to warn you about the risk of reaching the 6-byte RBA or LRSN limits. The increased size of all log records also accelerates progress toward the 6-byte RBA logging limit.

You must continuously monitor the RBA and LRSN values until all catalog, directory, and user objects are converted to the 10-byte RBA or LRSN format. Failure to convert page sets before the 6-byte soft logging limit is reached results in failed updates with reason code 00C2026D, and any objects still in the 6-byte format become read-only. RBA or LRSN values greater than x'F00000000000' indicate that your system is at risk of reaching the 6-byte logging limit.

For instructions, see Converting the BSDS to the 10-byte RBA and LRSN (Db2 Administration Guide).

Catalog table columns

The Db2 catalog and directory contain RBA and LRSN information in several tables.

The definition of the catalog and directory columns that contain RBA or LRSN values changes from 6-byte format to 10-byte format during the migration from CM to NFM. The catalog columns might be physically stored as either 6 bytes or 10 bytes. However, the values are converted to the 10-byte format as necessary when they are used in Db2. The actual values in the catalog depend on conversion step because the ENFM process is not a locked atomic operation.

The following situations are possible:

- In CM, the catalog and directory store 6-byte values.
- In ENFM, the format of the stored values depends on whether a specific table definition changed.

- In NFM and ENFM*, 10-byte values are stored.
- In CM*, the format depends on whether a specific table definition changed, if ENFM processing was not complete. If ENFM processing was complete at any point before the transition to CM*, 10-byte values are stored.

Some 6-byte values still exist until a REORG of the affected catalog and directory tables is complete. The 6-byte values are padded with zeros when they are retrieved.

For more information, see Db2 migration modes (Db2 Installation and Migration).

Shared communication area (SCA)

The SCA is used to track and communicate data pertinent to a data sharing group. This data always includes some LRSN and RBA values and there might be many such values, depending on the exception states of database objects.

The SCA is reformatted by a rebuild that is triggered during NFM migration. This step is not optional. In addition, an SCA rebuild might happen during group restart if the old-format SCA is found and the data sharing group is not in conversion mode.

After migration to NFM, the contents of some of the log records change to support the logging of SCA information after the SCA format change, regardless of the actual log format.

Utilities

In all migration modes, RBA and LRSN values are displayed in 10-byte format. This 10-byte display is unrelated to migration of the catalog or directory, conversion of individual objects to extended format, or BSDS conversion. For recovery purposes, this 10-byte format is the preferred input format for Db2. When 10-byte RBA or LRSN values are specified as input to Db2, the values are converted to 6-byte format internally, as needed. For more information, see Utility RBA and LRSN handling.

Work files

Data pages and space map pages for the work file database use the 10-byte format as soon as they are first accessed in Db2 11 (in any migration mode), regardless of whether the Db2 subsystem is migrated from DB2 10 or is a new installation. However, for migrated subsystems, the Db2 catalog is not updated to reflect the format of the work files. For more information about work files, see Work file database (Introduction to Db2 for z/OS).

When object and BSDS formats do not match

You can convert database objects to the 10-byte format any time that you are ready, after migration to Db2 11 new-function mode.



Attention: In Db2 subsystems that are not data sharing members, if Db2 is already at risk of reaching the 6-byte RBA limit, it is strongly recommended that you first convert all catalog and directory objects, then convert all user objects to the 10-byte RBA format, before you convert the BSDS.

In Db2 subsystems that are not data sharing members, always convert all Db2 catalog, directory, and user objects to use the extended 10-byte RBA format before you convert the BSDS, especially if Db2 is close to reaching the logging limit for the 6-byte RBA. Failure to convert page sets to the 10-byte RBA format before Db2 reaches the 6-byte logging limit results in failed updates with reason code 00C2026D. No updates are allowed for any object that is still in the 6-byte format.

You must continuously monitor the RBA and LRSN values until all catalog, directory, and user objects are converted to the 10-byte RBA or LRSN format. Failure to convert page sets before the 6-byte soft logging limit is reached results in failed updates with reason code 00C2026D, and any objects still in the 6-byte format become read-only. RBA or LRSN values greater than x'F00000000000' indicate that your system is at risk of reaching the 6-byte logging limit.

In data sharing, you do not need to convert all BSDSs in a data sharing group before the conversion of database objects begins. When an object is converted to the extended format, the change means that a 10-byte LRSN or RBA value is stored on the page, instead of a 6-byte value. That change does not mean that the converted object must have nonzero bytes in the extended area.

If an object is in basic format and the log uses the 10-byte format, the LRSN that is stored in PGLOGRBA is truncated to fit. If a database object is in the extended format, and the log remains in

the 6-byte format, LRSN values that are stored in the object are padded with zeros to the 10-byte format. Outside of data sharing environments, similar rules apply to RBA values.

When an object is in extended format and some members of the data sharing group have BSDS and logs in different formats, the order of updates is maintained. However, LRSN values from some members must be padded with zeros.

For a simple example, consider a data sharing group with two members:

- M10 is a member that has logs in the 10-byte format.
- M6 is a member that has logs in the 6-byte format.

Assume that the same data sharing group has two tables:

- TExt is a table with extended format.
- TBasic is a table with basic format.

The following illustration shows how a sequence of updates might look for the example data sharing group. These time values are for illustrative purposes. They are not representative of typical LRSN values because they correspond to updates that were completed in December, 1908.

```
      Time                Update       Content of PGLOGRBA or PGBigRBA
001000000001000002  M10 updates TBasic   100000000001 A
001000000001000003  M10 updates TExt   00100000000001000003 B
001000000001000004  M6  updates TExt   00100000000002000000 C
001000000001000005  M6  updates TBasic   100000000002 D
001000000002000001  M10 updates TBasic   100000000003 E
001000000002000003  M10 updates TExt   00100000000002000003 F
```

Each of the example updates can be from different transactions, and the last two must be from separate transactions.

The logic ensures that the PGLOGRBA or PGBigRBA does not decrease, even though the two subsystems are logging updates with different formats.

- For the first update A, the LRSN is truncated before it is placed in PGLOGRBA.
- The value for the second update B stores the 10-byte format with full precision.
- For the third update C, member M6 must generate an LRSN value that is greater than the 6 bytes that correspond to the old LRSN value. (If the LRSN is beyond the 6-byte range, updates are not allowed).
- For the fourth update D, member M6 again generates a value that is greater than the existing PGLOGRBA or PGBigRBAvalue.
- For the fifth update E, M10 must generate a larger value. A value greater than 00100000002FFFFFF is used because the TBasic table uses a 6-byte format LRSN.
- For the last update F, the only requirement is that the LRSN must be greater than the existing value, so the time of the log record is used.

The log entry for the fifth update E occurs later in the log for M10 (a higher RBA value) because it was delayed in generating the LRSN value. This situation requires that the last two updates are from different transactions. Otherwise, the sixth update must wait for the fifth update to complete to ensure that the sixth transaction has a later LRSN and a later sequence in the log.

Related concepts

Expanded RBA and LRSN log records

In Db2 11, the RBA and LRSN log records are expanded from basic 6-byte format to extended 10-byte format.

Related tasks

What to do before RBA or LRSN limits are reached (Db2 Administration Guide)

Related information

Reading log records (Db2 Administration Guide)

Db2 11 for z/OS Technical Overview (IBM Redbooks)

Chapter 17. Function that Db2 11 no longer supports

If you are migrating to Db2 11 from Db2 10, be aware of function that is no longer supported.

The following function is not supported in Db2 11.

Password protection for active log and archive log data sets

In Db2 11, password protection for active log and archive log data sets is no longer supported.

Previous NEWFUN values

In Db2 11, DSNHDECP no longer supports values of NEWFUN=V8 or NEWFUN=V9.

Some supplied routines

The following supplied routines are removed in Db2 11 and are unavailable to callers after migration to conversion mode. A report is added to the DSNTIJPM premigration job to detect occurrences of these routines on an existing subsystem or data sharing group, and to specify that these routines are not available in Db2 11.

- SYSPROC.DSNAEXP
- AMI-based Db2 MQ functions
  - DB2MQ1C.GETCOL
  - DB2MQ1C.MQPUBLISH
  - DB2MQ1C.MQREAD
  - DB2MQ1C.MQREADALL
  - DB2MQ1C.MQREADALLCLOB
  - DB2MQ1C.MQREADCLOB
  - DB2MQ1C.MQRECEIVE
  - DB2MQ1C.MQRECEIVEALL
  - DB2MQ1C.MQRECEIVEALLCLOB
  - DB2MQ1C.MQRECEIVECLOB
  - DB2MQ1C.MQSEND
  - DB2MQ1C.MQSUBSCRIBE
  - DB2MQ1C.MQUNSUBSCRIBE
  - DB2MQ2C.GETCOL
  - DB2MQ2C.MQPUBLISH
  - DB2MQ2C.MQREAD
  - DB2MQ2C.MQREADALL
  - DB2MQ2C.MQREADALLCLOB
  - DB2MQ2C.MQREADCLOB
  - DB2MQ2C.MQRECEIVE
  - DB2MQ2C.MQRECEIVEALL
  - DB2MQ2C.MQRECEIVEALLCLOB
  - DB2MQ2C.MQRECEIVECLOB
  - DB2MQ2C.MQSEND
  - DB2MQ2C.MQSUBSCRIBE

© Copyright IBM Corp. 2013, 2019

- DB2MQ2C.MQUNSUBSCRIBE
- DB2MQ1N.GETCOL
- DB2MQ1N.MQPUBLISH
- DB2MQ1N.MQREAD
- DB2MQ1N.MQREADALL
- DB2MQ1N.MQREADALLCLOB
- DB2MQ1N.MQREADCLOB
- DB2MQ1N.MQRECEIVE
- DB2MQ1N.MQRECEIVEALL
- DB2MQ1N.MQRECEIVEALLCLOB
- DB2MQ1N.MQRECEIVECLOB
- DB2MQ1N.MQSEND
- DB2MQ1N.MQSUBSCRIBE
- DB2MQ1N.MQUNSUBSCRIBE
- DB2MQ2N.GETCOL
- DB2MQ2N.MQPUBLISH
- DB2MQ2N.MQREAD
- DB2MQ2N.MQREADALL
- DB2MQ2N.MQREADALLCLOB
- DB2MQ2N.MQREADCLOB
- DB2MQ2N.MQRECEIVE
- DB2MQ2N.MQRECEIVEALL
- DB2MQ2N.MQRECEIVEALLCLOB
- DB2MQ2N.MQRECEIVECLOB
- DB2MQ2N.MQSEND
- DB2MQ2N.MQSUBSCRIBE
- DB2MQ2N.MQUNSUBSCRIBE

Some subsystem parameters

The following subsystem parameters are removed in Db2 11:

- ASSIST
- COORDNTR
- DISABSCL
- DISALLOW_DEFAULT_COLLID
- MSVGP
- MSVGP2
- OJPERFEH
- OPTIOPIN
- OPTIOWGT
- OPTIXIO
- PGRNGSCR
- PTCDIO
- RETVLCFK

- SEQCACH
- SEQPRES
- SMSDCFL
- SMSDCIX
- STATCLUS

An application programming default value

The following application programming default value is removed in Db2 11:

- CHARSET

BIND PACKAGE options ENABLE and DISABLE (REMOTE) REMOTE (*location-name*,...,<*luname*>,...)

In Db2 11, you cannot use the BIND PACKAGE options ENABLE and DISABLE (REMOTE) REMOTE (*location-name*,...,<*luname*>,...) to enable or disable specific remote connections. You can use the ENABLE(REMOTE) or DISABLE(REMOTE) options to enable or disable all remote connections.

Sysplex query parallelism

In Db2 11, Sysplex query parallelism is no longer supported. Packages that used Sysplex query parallelism in releases before Db2 11 use CPU parallelism in Db2 11.

DSN1CHKR utility

In Db2 11, the DSN1CHKR utility is no longer supported. The DSN1810I and DSN1816I messages are issued when the DSN1CHKR utility is invoked.

Related reference

Deprecated function in Db2 11
Certain capabilities that Db2 11 for z/OS supports are *deprecated*, meaning that their use is discouraged. Although they remain supported except as noted below in Db2 11, support is likely to be removed eventually.

Chapter 18. Deprecated function in Db2 11

Certain capabilities that Db2 11 for z/OS supports are *deprecated*, meaning that their use is discouraged. Although they remain supported except as noted below in Db2 11, support is likely to be removed eventually.

Avoid creating new dependencies that rely on deprecated function, and develop plans to remove any dependencies on such function.

Table 24. Deprecated functions in Db2 11

Deprecated function	Recommended alternative	Support removed
6-byte RBA and LRSN format for the BSDS	Convert the BSDS to use the extended 10-byte RBA and LRSN formats. The BSDS conversion must be completed before migration to Db2 12.	Db2 12
ALCUNIT subsystem parameter	Set ALCUNIT to CYL. Later Db2 releases always use the behavior of this setting.	Db2 12
Basic row format table spaces	Use reordered row format. Starting in Db2 12, any table space that uses basic row format is automatically converted to reordered row format when you run one of the following utilities: • LOAD REPLACE with the ROWFORMAT RRF option, or LOAD REPLACE without the ROWFORMAT option. The ROWFORMAT option is deprecated and will be removed eventually. • REORG TABLESPACE with the ROWFORMAT RRF option, or REORG TABLESPACE without the ROWFORMAT option. The ROWFORMAT option is deprecated and will be removed eventually.	—
BIND PLAN command MEMBER option	Use BIND PACKAGE commands to bind DBRMs into packages explicitly.	—
CACHEDYN_FREELOCAL subsystem parameter	Set the value to 0. Later Db2 releases always use the behavior of this setting.	Db2 12
CACHE_DEP_TRACK_STOR_LIM subsystem parameter	Starting in Db2 12, storage is allocated from the SCB pool.	Db2 12

© Copyright IBM Corp. 2013, 2019

Table 24. Deprecated functions in Db2 11 (continued)

Deprecated function	Recommended alternative	Support removed
COPY utility CHANGELIMIT option	Use the DSNACCOX stored procedure to determine if the object needs to be copied.	—
CATALOG (in DSN6ARVP) subsystem parameter	Set CATALOG (in DSN6ARVP) to YES. Later Db2 releases always use the behavior of this setting.	Db2 12
CHECK_SETCHKP subsystem parameter	Set CHECK_SETCHKP to NO. Later Db2 releases always use the behavior of this setting.	Db2 12
COMPRESS_SPT01 subsystem parameter	Set COMPRESS_SPT01 to YES. Later Db2 releases always use the behavior of this setting.	Db2 12
CONTSTOR subsystem parameter	Set CONTSTOR to NO. Later Db2 releases always use the behavior of this setting.	Db2 12
DB2SORT subsystem parameter	Set DB2SORT to ENABLE. Later Db2 releases always use the behavior of this setting.	Db2 12
DISALLOW_SEL_INTO_UNION subsystem parameter.	Modify applications to remove any use of UNION or UNION ALL as the outermost from-clause of a SELECT INTO statement. Then set DISALLOW_SEL_INTO_UNION to YES.	—
SYSPROC.DSNACCOR stored procedure	Use the SYSPROC.DSNACCOX stored procedure.	V12R1M504
DSNRLMT*xx* table format from before Db2 11	Use the current format for resource limit facility (RLF) tables. See Convert RLF tables to the current format (Db2 Installation and Migration).	—
DSNRLST*xx* table format from before DB2 Version 8	Use the current format for resource limit facility (RLF) tables. See Convert RLF tables to the current format (Db2 Installation and Migration).	Db2 12
SYSPROC.DSNTBIND stored procedure	Use the SYSPROC.ADMIN_COMMAND_DSN stored procedure.	—
SYSPROC.DSNUTILS stored procedure	Use the SYSPROC.DSNUTILU stored procedure.	—
DSNTPSMP SQL procedure processor	The SQL procedure processor, DSNTPSMP, is one of several methods that you can use to create and prepare an external SQL procedure, which are also deprecated.	—

Table 24. Deprecated functions in Db2 11 (continued)

Deprecated function	Recommended alternative	Support removed
SYSPROC.DSNWZP	Use the SYSPROC.ADMIN_INFO_SYSPARM stored procedure.	—
EDMPOOL subsystem parameter	Starting in Db2 12, all EDM storage is above the bar.	Db2 12
External SQL procedures	Use native SQL procedures, which are more fully supported, easier to maintain, and typically perform better that external SQL procedures. For more information, see Migrating an external SQL procedure to a native SQL procedure (Db2 Application programming and SQL)	—
Hash-organized tables	Alter tables to drop hash organization, and , create indexes to support fast index traversal in Db2 12.	Beginning in Db2 12 with application compatibility level V12R1M504, Db2 no longer supports creating hash-organized tables or altering tables to use hash-organization. Existing hash organized tables remain supported. However, that support is likely to be removed in the future.
IFCID 0366	Use IFCID 0376.	Db2 12
INDEX_IO_PARALLELISM subsystem parameter	Set INDEX_IO_PARALLELISM to YES. Later Db2 releases always use the behavior of this setting.	Db2 12
LEMAX subsystem parameter	Starting in Db2 11, the LEMAX value is not used.	Db2 12
LOAD utility IDENTITYOVERRIDE option	Use the OVERRIDE(IDENTITY) option.	—
LOAD utility PERIODOVERRIDE option	Use the OVERRIDE(SYSTEMPERIOD) option.	—
LOAD utility TRANSIDOVERRIDE option	Use the OVERRIDE(TRANSID) option.	—
LOBVALA subsystem parameter	Use the default value. Starting in Db2 12, the database manager automatically determines the amount of storage for processing LOB values.	Db2 12
LOBVALS subsystem parameter	Use the default value. Starting in Db2 12, the database manager automatically determines the amount of storage for processing LOB values.	Db2 12

Table 24. Deprecated functions in Db2 11 (continued)

Deprecated function	Recommended alternative	Support removed
MATERIALIZE_NODET_SQLTUDF subsystem parameter	Set MATERIALIZE_NODET_SQLTUDF to YES. Later Db2 releases always use the behavior of this setting.	—
MINSTOR subsystem parameter	Set MINSTOR to NO. Later Db2 releases always use the behavior of this setting.	Db2 12
NEWFUN *dsnhdecp* parameter	Use NEWFUN=V*nn* instead of YES or NO values. Starting in Db2 12, use SQLLEVEL.	Db2 12
NEWFUN SQL processing option	Use NEWFUN(*nn*) instead of YES or NO values. Starting in Db2 12, use SQLLEVEL. NEWFUN is ignored if SQLLEVEL is specified.	—
OBJECT_CREATE_FORMAT subsystem parameter.	Set this subsystem parameter value to NOBASIC before migrating to Db2 12. Later Db2 releases always use the behavior of this setting.	Db2 12
ODBC 2.0 function	See Deprecated ODBC functions (Db2 Programming for ODBC).	—
REORG_IGNORE_FREESPACE subsystem parameter	Set REORG_IGNORE_FREESPACE to NO. Later Db2 releases always use the behavior of this setting.	Db2 12
REORG INDEX utility LAEFDISTLIMIT and REPORT only options	Use the DSNACCOX stored procedure to determine whether the object needs to be reorganized	—
REORG INDEX utility UNLOAD ONLY option	Use the UNLOAD utility.	—
REORG INDEX utility UNLOAD PAUSE option	Use the DIAGNOSE utility to stop the process.	—
REORG TABLESPACE utility UNLOAD EXTERNAL option	Use the UNLOAD utility.	—
REORG TABLESPACE utility INDREFLIMIT and REPORTONLY options	Use the DSNACCOX stored procedure to determine whether the object needs to be reorganized.	—
REORG TABLESPACE utility OFFPOSLIMIT and REPORTONLY options	Use the DSNACCOX stored procedure to determine whether the object needs to be reorganized.	—
REORG TABLESPACE utility PARALLEL option	Use the LISTPARTS option instead.	—
REORG TABLESPACE utility UNLOAD ONLY option	Use the UNLOAD utility.	—

Table 24. Deprecated functions in Db2 11 (continued)

Deprecated function	Recommended alternative	Support removed
REORG TABLESPACE utility UNLOAD PAUSE option	Use the UNLOAD utility FORMAT INTERNAL option.	—
RBALRSN_CONVERSION keyword of the LOAD, REORG, and REBUILD INDEX utilities	Starting in Db2 12, utilities that specify RBALRSN_CONVERSION(BASIC) always fail. If the target objects is in the 6-byte page format, utilities that specify RBALRSN_CONVERSION(NONE) also fail.	Db2 12
REPAIR VERSIONS utility	Use the REPAIR CATALOG utility.	—
RRF subsystem parameter	Set RRF to YES. Later Db2 releases always use the behavior of this setting.	Db2 12
SPT01_INLINE_LENGTH subsystem parameter	Set SPT01_INLINE_LENGTH to 32138.Later Db2 releases always use the behavior of this setting.	Db2 12
DB2XML.SOAPHTTPC supplied user-defined function	Use the DB2XML.SOAPHTTPNC user-defined function.	—
DB2XML.SOAPHTTPV supplied user-defined function	Use the DB2XML.SOAPHTTPNV user-defined function.	—
SQWIDSC subsystem parameter	Set SQWIDSC to YES. Later Db2 releases always use the behavior of this setting.	Db2 12
Synonyms	Use aliases when writing new SQL statements or creating portable applications. Aliases behave the same for the Db2 family of products.	—
SYSIBM.SYSROUTINES_OPTS catalog table	This catalog table supports the DSNTPSMP SQL procedure processor for creating and preparing external SQL procedures, which are also deprecated.	—
SYSIBM.SYSROUTINES_SRC catalog table	This catalog table supports the DSNTPSMP SQL procedure processor for creating and preparing external SQL procedures, which are also deprecated.	—

Table 24. Deprecated functions in Db2 11 (continued)

Deprecated function	Recommended alternative	Support removed
Non-UTS table spaces for base tables, including segmented (non-UTS), partitioned (non-UTS), and simple table spaces.	Use partition-by-growth or partition-by-range universal table spaces instead. In Db2 12, Packages bound with APPLCOPMAT(V12R1M504) or higher cannot create objects of the following types: • New partitioned (non-UTS) table spaces • New segmented (non-UTS) table spaces • New tables in existing segmented (non-UTS) table spaces • New tables in existing simple table spaces You cannot create new simple table spaces in any supported Db2 release.	—
TEMP database	Db2 uses the work file database instead.	—
UTILITY_OBJECT_CONVERSION subsystem parameter	Set UTILITY_OBJECT_CONVERSION to EXTENDED before migrating to Db2 12. Later Db2 releases always use the behavior of this setting.	Db2 12
UTSORTAL subsystem parameter	Set UTSORTAL to YES. Later Db2 releases always use the behavior of this setting.	Db2 12
XMLVALA subsystem parameter	Use the default value. In Db2 12, the database manager automatically determines the amount of storage for processing XML values.	Db2 12
XMLVALS subsystem parameter	Use the default value. In Db2 12, the database manager automatically determines the amount of storage for processing XML values.	Db2 12

Related concepts

Incompatible changes for migration to Db2 11
Before migrating your Db2 10 subsystem to Db2 11, familiarize yourself with incompatible changes that might impact your migration. Plan to resolve any such applicable incompatible changes that apply to your Db2 environment before or during the Db2 11 migration process.

Related tasks

Preparing your system to install or migrate to Db2 11 (Db2 Installation and Migration)

Information resources for Db2 11 for z/OS and related products

Information about Db2 11 for z/OS and products that you might use in conjunction with Db2 11 is available online in IBM Knowledge Center.

You can find the complete set of product documentation for Db2 11 for z/OS in IBM Knowledge Center.

You can also download other PDF format manuals for Db2 11 for z/OS from IBM Knowledge Center in PDF format manuals for Db2 11 for z/OS (Db2 Home).

© Copyright IBM Corp. 2013, 2019

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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

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

IBM Director of Licensing IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785 US

For license inquiries regarding double-byte (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.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

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 jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

IBM Director of Licensing IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785 US

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.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

© Copyright IBM Corp. 2013, 2019

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

Programming interface information

This information is intended to help you to learn about and plan to use Db2 11 for z/OS. This information also documents General-use Programming Interface and Associated Guidance Information and Product-sensitive Programming Interface and Associated Guidance Information provided by Db2 11 for z/OS.

General-use Programming Interface and Associated Guidance Information

General-use Programming Interfaces allow the customer to write programs that obtain the services of Db2 11 for z/OS.

General-use Programming Interface and Associated Guidance Information is identified where it occurs by the following markings:

GUPI General-use Programming Interface and Associated Guidance Information... GUPI

Product-sensitive Programming Interface and Associated Guidance Information

Product-sensitive Programming Interfaces allow the customer installation to perform tasks such as diagnosing, modifying, monitoring, repairing, tailoring, or tuning of this IBM software product. Use of such interfaces creates dependencies on the detailed design or implementation of the IBM software product. Product-sensitive Programming Interfaces should be used only for these specialized purposes. Because of their dependencies on detailed design and implementation, it is to be expected that programs written to such interfaces may need to be changed in order to run with new product releases or versions, or as a result of service.

Product-sensitive Programming Interface and Associated Guidance Information is identified where it occurs by the following markings:

PSPI Product-sensitive Programming Interface and Associated Guidance Information... PSPI

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered marks 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: http://www.ibm.com/legal/copytrade.shtml.

Linux® is a registered trademark of Linus Torvalds in the United States, other countries, or both.

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

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

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Terms and conditions for product documentation

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.

Privacy policy considerations

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM’s Privacy Policy at http://www.ibm.com/privacy and IBM’s Online Privacy Statement at http://www.ibm.com/privacy/details the section entitled “Cookies, Web Beacons and Other Technologies” and the “IBM Software Products and Software-as-a-Service Privacy Statement” at http://www.ibm.com/software/info/product-privacy.

Glossary

The glossary is available in IBM Knowledge Center.

See the Glossary topic for definitions of Db2 for z/OS terms.

© Copyright IBM Corp. 2013, 2019

Index

Numerics

10-byte format
  extended format
    RBA and LRSN 111
  RBA and LRSN 111
64-bit JVM 37

A

ACCESS DATABASE command 31
access paths
  resetting statistics 31
ALTER TABLESPACE
  PCTFREE FOR UPDATE 20
ALTER TABLESPACE statement
  PCTFREE FOR UPDATE option 20
altering limit keys
  improved availability 24
  materialized query tables 30
APPLCOMPAT 61
application compatibility 44
application programs
  compatibility 61
application release incompatibilities 47
applications
  new 35
archive tables
  overview 30
archive-enabled tables
  overview 30
arrays
  functions 38
  routines 38
  stored procedures 38
ASUTIME
  and dynamic SQL statements 47
automatic rebinds
  old plans and packages 48
availability 23
availability improvements
  compression dictionary 26
  Db2 support for GDPS Continuous Availability with zero data loss 26
  index recovery 23
  online schema enhancements 23

B

BIND QUERY command 21
buffer pools 14

C

CAST 55
catalog tables
catalog tables *(continued)*
  SYSIBM.SYSSTATFEEDBACK 31
catalogs
  changes 93
column masking
  improvements 39
command release incompatibilities 70
commands
  ACCESS DATABASE 31
  BIND QUERY 21
  changes 79
continuous block fetch
  performance improvements 19
CPU reductions
  distributed applications 16
CREATE TABLESPACE statement
  PCTFREE FOR UPDATE option 20
CREATE TRIGGER 59
cross-loader function
  improvements 37

D

data validation
  DSN1COPY 29
Db2 enhancements after general availability 3
declared global temporary tables
  not logged 38
directory
  recovery improvements 16
distributed applications
  CPU reductions 16
DROP PENDING CHANGES
  improvements 30
DSN_PREDICAT_TABLE table
  table 21
DSN_PREDICATE_SELECTIVITY table 21
DSN_STAT_FEEDBACK table 31
DSN1COPY
  data validation 29
  improvements 29
duplicate removal
  performance improvements 18
dynamic SQL statements
  change to ASUTIME determination 47

E

ease of use 29
enhancements
  authorization checking 17, 33
EXPLAIN tables
  changes 99
  DSN_PREDICAT_TABLE 21
  DSN_PREDICATE_SELECTIVITY 21
  DSN_STAT_FEEDBACK 31
  encoding 99

EXPLAIN tables *(continued)*
Version 10 format 99

F

filter factors
access path selection
influencing 21
overriding 21
free space
managing for UPDATE operations 20
functions
array parameters 38
column 87
new 87
scalar 87

G

general-use programming information, described 128
group buffer pool
storage relief 17
group-buffer-pool write-around protocol 17
GUPI symbols 128

I

IFCID (instrumentation facility component identifier)
changed 101
new 101
IFCID 0366 61
improvements
overview 1
incompatibilities of releases
applications and SQL 47
commands 70
other 72
storage 72
unsupported functions 115
utilities 65
index cleanup
automatic 18
indexes
recovering
GRECP status 23
LPL status 23
REBUILD-pending status 23
installation
improvements 43
installation verification procedure (IVP)
improvements 43
introduction 1
IVP (installation verification procedure)
improvements 43

J

Java
stored procedures 37

L

LIKE blank insignificant 38
LIKE blank significant 38
LIKE predicate
enhancements 38
LIKE blank insignificant 38
LIKE blank significant 38
limit keys
altering 24
limited block fetch
ODBC applications 13
links
non-IBM Web sites 128
LOAD utility
INCURSOR option
improvements 37
XML support 37
PARALLEL 14
parallelism improvements 14
performance improvements 14
log records 26
log writes
performance improvements 17
synchronous 17
LRSN
expanded 19
extended 111

M

materialized query tables
altering limit keys 30
migration
considerations 47
improved 43
planning 43
monitoring improvements
work file database 25

N

not logged
declared global temporary tables 38

O

ODBC
variable timestamp data 37
ODBC applications
performance improvements 13
online alter limit key 24
online REORG
improvements 25
optimization
statistics feedback 31

P

packages
automatic rebinds 48
parallel processing 25
PCTFREE FOR UPDATE option
ALTER TABLESPACE statement 20
CREATE TABLESPACE statement 20
pending definition changes

pending definition changes *(continued)*
recovery prior to materialization 23
removing 30
performance improvements
continuous block fetch 19
duplicate removal 18
indexes 13
LOAD utility 14
ODBC applications 13
partitions 16
REORG TABLESPACE 15
ROLLBACK TO SAVEPOINT statement 16
stored procedure calls 13
persistent threads
long-running
performance improvements 15
plans
automatic rebinds 48
point-in-time recovery
prior to materialization of pending definition changes 23
predicates
added 20
overriding selectivity 21
removed 20
product-sensitive programming information, described 128
profiles
special registers 32
programming interface information, described 128
pseudo-deleted index entries
cleanup 18
pseudo-empty index pages
cleanup 18
PSPI symbols 128

Q

queries
transformations 20

R

RBA
expanded 19
extended 111
real-time statistics
externalizing immediately 31
rebinds
automatic 48
REBUILD INDEX utility 31
recovery
performance improvements 16
regulatory compliance 33
release incompatibilities
applications and SQL 47
commands 70
other 72
storage 72
unsupported functions 115
utilities 65
RELEASE(DEALLOCATE) behavior
performance improvements 15
removed functions 115
REORG INDEX utility 31
REORG TABLESPACE utility
MAPPINGDATABASE keyword 25
REORG_MAPPING_DATABASE subsystem parameter 25
REPAIR
CATALOG 29
ROLLBACK TO SAVEPOINT statement
performance improvements 16
routines
array parameters 38
RRSAF functions 39
RUNSTATS utility 31

S

samples
improvements 43
security 33
selectivity
catalog tables
SYSIBM.SYSQUERY 21
SYSIBM.SYSQUERYPREDICATE 21
SYSIBM.SYSQUERYSEL catalog table 21
overriding for predicates 21
SQL path
SYSIBMADM 55
SQL release incompatibilities 47
SQL statements
ALTER TABLESPACE
PCTFREE FOR UPDATE 20
changes 87
clauses
changed 87
new 87
new 87
SQL variables
arrays 38
statistics
access path
resetting 31
collection improvements 31
missing and conflicting
identifying 31
real-time
externalizing immediately 31
storage relief
group buffer pool 17
stored procedures
array parameters 38
Java 37
subsystem parameters
changes 105
PCTFREE_UPD
subsystem parameter 20
SYSIBM.SYSQUERY catalog table 21
SYSIBM.SYSQUERYPREDICATE catalog table 21
SYSIBM.SYSQUERYSEL catalog table 21
SYSIBM.SYSSTATFEEDBACK catalog table 31
SYSIBMADM schema 55
SYSIBMADM.GET_ARCHIVE
overview 30
SYSIBMADM.MOVE_TO_ARCHIVE
overview 30

T

temporal auditing 40
temporal support
  enhancements 35
trace enhancements 101
trace field QW0366FN 61
trace field QW0376FN 61
transformations
  queries 20

U

unsupported functions 115
UPDATE operations
  reserving free space 20
user-defined global variables 38
utilities
  changes 83
  LOAD 31
  parallel processing 25
  REBUILD INDEX 31
  release incompatibilities 65
  REORG INDEX 31
  REORG TABLESPACE 20, 31
  RUNSTATS 31

V

variable timestamp data
  ODBC 37
views
  period clauses 35
  period specifications 35

W

WHEN clause search-condition references system-period
temporal table 59
work file database
  monitoring improvements 25

X

XML
  data type 37
XSLTRANSFORM 39

Z

zIIP
  index cleanup 18



Product Number: 5615-DB2
5697-P43

GC19-4068-11

