

Db2 13 for z/OS

What's New?

Last updated: 2024-03-12



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 [IBM Documentation](#).

2024-03-12 edition

This edition applies to Db2® 13 for z/OS® (product number 5698-DB2®), Db2 13 for z/OS Value Unit Edition (product number 5698-DBV), 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 2022.**

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

Contents

About this information.....	V
Terminology and citations.....	vi
How to send comments.....	vi
 Part 1. What's new in Db2 13.....	 1
Chapter 1. Overview of what's new in Db2 13.....	3
Chapter 2. New-function APARs for Db2 13.....	33
New-function APARs for Db2 13 in 2024.....	36
New-function APARs for Db2 13 in 2023.....	37
New-function APARs for Db2 13 in 2022.....	47
Chapter 3. Db2 13 function levels.....	55
Function level 504 (APAR PH54919 - October 2023).....	55
Function level 503 (APAR PH51469 - February 2023).....	60
Function level 502 (APAR PH47891 - October 2022).....	63
Function level 501 (Db2 13 installation or migration - May 2022).....	67
Function level 500 (for migrating to Db2 13 - May 2022).....	71
Function level 100 (for migrating to Db2 13 - May 2022).....	78
 Part 2. Changes to plan for in Db2 13.....	 87
Chapter 4. Incompatible changes in Db2 13.....	89
Chapter 5. Storage changes in Db2 13.....	93
Chapter 6. Command changes in Db2 13.....	95
Chapter 7. SQL changes in Db2 13.....	99
Chapter 8. Utility changes in Db2 13.....	103
Chapter 9. Catalog changes in Db2 13.....	105
Chapter 10. EXPLAIN table changes.....	109
Chapter 11. IFCID changes in Db2 13.....	111
Chapter 12. Subsystem parameter changes in Db2 13.....	117
Chapter 13. The extended 10-byte RBA and LRSN in Db2 13.....	123
Chapter 14. Function that Db2 13 no longer supports.....	127
Chapter 15. Deprecated function in Db2 13.....	129
 Part 3. Adopting new capabilities in Db2 13 continuous delivery.....	 133
Chapter 16. Function levels and related levels in Db2 13.....	135

Chapter 17. Determining the Db2 code level, catalog level, and function level.....	139
Chapter 18. Testing Db2 function level activation.....	143
Chapter 19. Identifying applications that are incompatible with catalog updates.....	145
Chapter 20. Activating Db2 13 function levels.....	147
Chapter 21. Activating Db2 function levels by using z/OSMF.....	151
Job DSNTIJOZ: bring Db2 subsystem parameter changes online.....	152
Chapter 22. Updating Db2 initialization parameters for function level activation.....	153
Chapter 23. Controlling the Db2 application compatibility level.....	155
Chapter 24. Responding to problems after function level activation.....	157
Chapter 25. How Db2 function levels are documented.....	159
Information resources for Db2 for z/OS and related products.....	161
Notices.....	163
Programming interface information.....	164
Trademarks.....	164
Terms and conditions for product documentation.....	164
Privacy policy considerations.....	165
Glossary.....	167
Index.....	169

About this information

This information provides an executive overview of new function in Db2 13 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 13 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 13 for z/OS, always use [IBM® Documentation](#) or download the latest PDF file from [PDF format manuals for Db2 13 for z/OS \(Db2 for z/OS in IBM Documentation\)](#).

Most documentation topics for Db2 13 for z/OS assume that the highest available function level is activated and that your applications are running with the highest available application compatibility level, with the following exceptions:

- The following documentation sections describe the Db2 13 migration process and how to activate new capabilities in function levels:
 - [Migrating to Db2 13 \(Db2 Installation and Migration\)](#)
 - [Part 1, "What's new in Db2 13," on page 1](#)
 - [Part 3, "Adopting new capabilities in Db2 13 continuous delivery," on page 133](#)
- [FL 500](#) A label like this one usually marks documentation changed for function level 500 or higher, with a link to the description of the function level that introduces the change in Db2 13. For more information, see [Chapter 25, "How Db2 function levels are documented," on page 159](#).

The availability of new function in Db2 13 depends on the type of enhancement, the activated function level, and the application compatibility levels of the applications. For a list of all available function levels in Db2 13, see [Chapter 3, "Db2 13 function levels," on page 55](#).

Function level 100

Db2 starts at function level 100 (V13R1M100) during migration to Db2 13, and fallback and coexistence with Db2 12 in data sharing remain possible. Many new capabilities in Db2 13 remain unavailable. For more information, see ["Function level 100 \(for migrating to Db2 13 - May 2022\)" on page 78](#).

Function level 500

Activating function level 500 (V13R1M500) prevents coexistence with and fallback to Db2 12. Function level 500 is also the first opportunity for applications to use many of the new capabilities in Db2 13. However, new capabilities that depend on Db2 13 catalog changes remain unavailable. For more information, see ["Function level 500 \(for migrating to Db2 13 - May 2022\)" on page 71](#).

Function level 501

Function level 501 (V13R1M501) is the first opportunity after migration to Db2 13 for applications to use new features and capabilities that depend on catalog changes in Db2 13. For more information, see ["Function level 501 \(Db2 13 installation or migration - May 2022\)" on page 67](#).

Some virtual storage and optimization enhancements take effect in function level 100. Optimization enhancements become available after full prepare of the SQL statements, depending on the statement type:

- For static SQL statements, after bind or rebind of the package.
- For non-stabilized dynamic SQL statements, immediately, unless the statement is in the dynamic statement cache.

- For stabilized dynamic SQL statements, after invalidation, free, or changed application compatibility level.

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.

The following terms are used as indicated:

Db2

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

IBM OMEGAMON® for Db2 Performance Expert on z/OS

Refers to any of the following products:

- IBM IBM OMEGAMON 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 about Db2 for z/OS documentation

Your feedback helps IBM to provide quality documentation.

Send any comments about Db2 for z/OS and related product documentation by email to db2zinfo@us.ibm.com.

To help us respond to your comment, include the following information in your email:

- The product name and version
- The address (URL) of the page, for comments about online documentation
- The book name and publication date, for comments about PDF manuals
- The topic or section title
- The specific text that you are commenting about and your comment

Related concepts

[About Db2 13 for z/OS product documentation \(Db2 for z/OS in IBM Documentation\)](#)

Related reference

[PDF format manuals for Db2 13 for z/OS \(Db2 for z/OS in IBM Documentation\)](#)

Part 1. What's new in Db2 13

Db2 13 for z/OS brings leading-edge innovation to reinforce Db2 for z/OS as a foundation for enterprise computing within the hybrid cloud world.

Db2 13 for z/OS delivers significant advances in all critical enterprise database success factors: availability, scalability, performance, security, and ease of use. The value of Db2 13 is also maximized by synergy with surrounding tools and technology. The latest advances in IBM zSystems hardware, new and improved development tooling, and add-on capabilities such as AI infusion and IBM Db2 Analytics Accelerator query acceleration, complement the new capabilities in Db2 13 to provide the most complete, intelligent and intuitive database environment yet.

Continuous delivery in Db2 13

Db2 13 evolves the *continuous delivery* of new capabilities and enhancements in a single service stream as soon as they are ready, which was introduced in Db2 12. The result is that you can benefit from new capabilities and enhancements without waiting for an entire new release. *Function levels* enable you to control the timing of the activation and adoption of new features, with the option to continue to apply corrective and preventative service without adopting new feature function.

New for Db2 13, function level 501 is available immediately at GA, and it activates any new capabilities that require catalog changes in Db2 13. For more information, and a list of all available function levels in Db2 13, see [Chapter 3, “Db2 13 function levels,” on page 55](#).

Tip: For the most current information, view documentation for Db2 continuous delivery and function levels in [IBM Documentation](#), or download the latest PDF edition from [PDF format manuals for Db2 13 for z/OS \(Db2 for z/OS in IBM Documentation\)](#).

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!](#)

Related information

[IBM Db2 13 for z/OS and More \(IBM Redbooks\)](#)

Chapter 1. Overview of what's new in Db2 13

Db2 13 introduces new capabilities and enhancements for simplified migration, SQL enhancements, applications management, IBM zSystems hardware synergy, availability and scalability, performance, and more!

Many of the new capabilities in Db2 13 take effect when you activate function level 500 or higher in Db2 13. The function level for each new capability is identified here.

Tip: This topic contains a comprehensive list of all new capabilities and enhancements available in Db2 13. It is most useful if you are preparing to install or migrate to Db2 13 for the first time. If you are looking for descriptions of the most recent new-function APARs, or new capabilities in specific Db2 13 function levels, see these topics instead:

- [Chapter 2, “New-function APARs for Db2 13,” on page 33](#)
- [Chapter 3, “Db2 13 function levels,” on page 55](#)

Tip: New capabilities and enhancements from many new-function APARs in Db2 12 are built-in when you migrate to Db2 13. If you do not apply the PTFs in Db2 12, plan for the changes to take effect when you migrate to Db2 13. See the APARs with availability dates earlier than 2022-06 in [New-function APARs for Db2 12](#).

Contents

[Simplified migration](#)

[SQL enhancements and applications management](#)

[IBM zSystems hardware synergy](#)

[Availability](#)

[Scalability](#)

[Performance](#)

[Db2 Utilities](#)

[Insight, instrumentation, and serviceability](#)

[“Hardware and software requirements for Db2 13” on page 30](#)

Simplified migration to Db2 13

Availability of Db2 13 new function

Migrations to Db2 13 for z/OS use a single phase. As in migrations to Db2 12, you use function levels to control the availability of most new function in Db2 13. For a complete list of available function levels, see [Chapter 3, “Db2 13 function levels,” on page 55](#).

At general availability, Db2 13 includes the following function levels:

Function level 100

Db2 starts at function level 100 (V13R1M100) during migration to Db2 13, and fallback and coexistence with Db2 12 in data sharing remain possible. Many new capabilities in Db2 13 remain unavailable. For more information, see [“Function level 100 \(for migrating to Db2 13 - May 2022\)” on page 78](#).

Function level 500

Activating function level 500 (V13R1M500) prevents coexistence with and fallback to Db2 12. Function level 500 is also the first opportunity for applications to use many of the new capabilities in Db2 13. However, new capabilities that depend on Db2 13 catalog changes remain unavailable. For more information, see [“Function level 500 \(for migrating to Db2 13 - May 2022\)” on page 71](#).

Function level 501

Function level 501 (V13R1M501) is the first opportunity after migration to Db2 13 for applications to use new features and capabilities that depend on catalog changes in Db2 13. For more information, see [“Function level 501 \(Db2 13 installation or migration - May 2022\)”](#) on page 67.

For more information, see [Activating Db2 13 function level 500 or higher \(Db2 Installation and Migration\)](#).

Simplified catalog migration

When migrating to Db2 13, you use the CATMAINT utility invoked by job DSNTIJTC to control the timing of the migration as in earlier releases, but it does not make any structural changes to the Db2 catalog. After the V13R1M100 CATMAINT completes, Db2 is at catalog level and function level V13R1M100. This is a change from Db2 12, which started with catalog level V12R1M500 at function level 100.

The first changes to catalog objects are delayed until you tailor the Db2 catalog for activation of function level 501.

For more information, see [Migrating your Db2 subsystem to Db2 13 \(Db2 Installation and Migration\)](#).

Subsystem parameter simplification

Function level 100 introduces changes to the default values for various subsystem parameters to match current best practices. It also removes a number of obsolete subsystem parameters. For a list of these changes, see [Chapter 12, “Subsystem parameter changes in Db2 13,”](#) on page 117.

Migration readiness reported in DISPLAY GROUP command output

Starting in Db2 12 with APAR PH50072 (June 2023), you can determine whether your Db2 12 data sharing group or standalone Db2 subsystem is ready for migration to Db2 13 by issuing a DISPLAY GROUP command with the DETAIL keyword.

Before this APAR, it can be difficult to determine whether all Db2 data sharing members are at sufficient code level and function level, and have the fallback SPE APAR applied.

The new migration readiness report in the DISPLAY GROUP output now indicates whether your environment is ready for migration to the next Db2 release. If Db2 is not ready for migration, the report also includes reason information.

For example, the following migration readiness report indicates that a Db2 12 subsystem has a sufficient code level and the fallback SPE is applied, but it is not ready to migrate to Db2 13 because the highest activated function level is not the last Db2 12 function level:

```
DSN7100I  -DB2A DSN7GCM
*** BEGIN DISPLAY OF GROUP(.....) CATALOG LEVEL(V12R1M500)
                                CURRENT FUNCTION LEVEL(V12R1M100)
                                HIGHEST ACTIVATED FUNCTION LEVEL(V12R1M100)
                                HIGHEST POSSIBLE FUNCTION LEVEL(V12R1M501)
                                PROTOCOL LEVEL(2)
                                GROUP ATTACH NAME(....)

-----
DB2      SUB      DB2      SYSTEM      IRLM
MEMBER   ID      SYS      CMDPREF   STATUS   LVL      NAME      SUBSYS   PRLMPR21
-----
.....   0 DB2A  -DB2A      ACTIVE   121510  UTEC6      PR21      PRLMPR21
-----

MIGRATION READINESS REPORT
-----
DB2      CODE      SPE      MIGRATION
MEMBER   LEVEL     APAR     ELIGIBLE
-----
.....   V12R1M510  PH37108  YES
-----

MIGRATION READINESS STATUS: SUBSYSTEM IS NOT READY FOR DB2 13
REASON: HIGHEST ACTIVATED FUNCTION LEVEL NOT V12R1M510
-----
```

The GET_CONFIG stored procedure is also updated to display the migration readiness information.

FL 504 Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

Tip: A follow-on APAR, PH58761 improves how the migration readiness status is displayed for Db2 13 members in coexistence situations. For more information, see [PH58761](#).

For more information, see the following related topics:

- [Check readiness for migration to Db2 13 \(Db2 Installation and Migration\)](#)
- [-DISPLAY GROUP \(Db2\) \(Db2 Commands\)](#)
- [DSN7100I \(Db2 Messages\)](#)
- [GET_CONFIG stored procedure \(Db2 SQL\)](#)
- [PH50072](#)

Specify MQ administration IDs in DSNTIJRT/DSNTRIN

Starting in Db2 13 function level V13R1M100 or higher with APAR PH55025 (September 2023), you can specify a list of authorization IDs to be granted ALL privileges on the Db2 MQ tables SYSIBM.MQPOLICY_TABLE and SYSIBM.MQSERVICE_TABLE when you install or migrate Db2. A new optional configuration (DB2OPT) keyword called MQ_ADMIN_ID is added to installation job DSNTIJRT for this purpose. A new MQ ADMIN ID(s) field is also added on panel DNSTIPG1 for setting this parameter. This change helps to support appropriate separation of duties between users of MQ functions and administrators of MQ policies and services.

Before this APAR, when DSNTIJRT/DSNTRIN generates the access list for the two MQ tables based on all the authorization IDs specified in the GRANTTO parameter of each Db2 MQ function, and the result is that any user with EXECUTE privilege on any Db2 MQ function also has ALL privileges on the two Db2 MQ tables. Thus, any user that is allowed to read from or write messages into MQ queues through Db2 MQ functions is also able to delete or change entries in SYSIBM.MQPOLICY_TABLE and SYSIBM.MQSERVICE_TABLE.

This new DSNTIJRT/DSNTRIN optional configuration DB2OPT parameter serves two purposes when specified:

- Specifying a list of authorization IDs that will be granted ALL privileges on the Db2 MQ tables.
- Specifying that only the SELECT privilege on the MQ tables will be granted to all users with EXECUTE privilege on any of the Db2 MQ functions.

The following example shows how to specify previous behavior in MQ_ADMIN_ID parameter setting in job DSNTIJRT:

```
//DSNTRIN EXEC PGM=DSNTRIN,COND=(4,LT),
//          PARM=('Db2SSN(!DSN!) MODE(INSTALL)',
//          ' AUTHID(!AUTHID!) SECDEFID(!SECDEFID!)',
//          ' DEFPKOWN(!DEFPKOWN!)')
//
//DB2OPT DD *
//STOGROUP(SYSDEFLT)
//INDEXSTOG(SYSDEFLT)
//BP4K(BP0)
//BP8K(BP8K0)
//BP16K(BP16K0)
//BP32K(BP32K)
//LOBBP8K(BP8K0)
//LOBBP16K(BP16K0)
//LOBBP32K(BP32K)
//IMS_SECURITY(Db2)
//MQ_SECURITY(Db2)
//SOAP_SECURITY(Db2)
//RTN_PKG_APPLCOMPAT(DEFAULT)
//RTN_PKG_PLANMGMT(DEFAULT)
//MQ_ADMIN_ID(DEFAULT)
```

The following example shows how to specify a list of authorization IDs in the MQ_ADMIN_ID parameter to get the new behavior:

```
MQ_ADMIN_ID(MQUID1,MQUID2,MQUID3)
```

For more information, see the following related topics:

- [Job DSNTIJRT \(Db2 Installation and Migration\)](#)
- [MQ ADMIN ID\(S\) field \(Db2 Installation and Migration\)](#)
- [DSNTIPG1: Installation preferences panel 2 \(Db2 Installation and Migration\)](#)
- [Db2 MQ tables \(Db2 Application programming and SQL\)](#)
- [PH55025](#)

SQL enhancements and applications management in Db2 13

Increased control for applications over how long to wait for a lock

Function level 500 introduces the CURRENT LOCK TIMEOUT special register and the SET CURRENT LOCK TIMEOUT SQL statement to allow the lock timeout value to be set at the application level. So, you can set a lock timeout interval that suits the needs of a specific application, or even an individual SQL statement. Doing so minimizes application lock contention and simplifies portability of applications to Db2, without the need to assign the application to a separate Db2 subsystem.

The value of the CURRENT LOCK TIMEOUT special register overrides the value of the IRLMRWT subsystem parameter. It applies to certain processes related to locking, like the claim or drain of an object and cached dynamic statement quiescing.

For more information, see [CURRENT LOCK TIMEOUT special register \(Db2 SQL\)](#) and [SET CURRENT LOCK TIMEOUT \(Db2 SQL\)](#).

You can limit use of CURRENT LOCK TIMEOUT by setting the new SPREG_LOCK_TIMEOUT_MAX subsystem parameter. For more information, see [LOCK TIMEOUT MAX \(SPREG_LOCK_TIMEOUT_MAX subsystem parameter\) \(Db2 Installation and Migration\)](#).

You can also use Db2 profile tables to specify an assignment for the CURRENT LOCK TIMEOUT special register, for both remote and local threads. See [Setting special registers by using profile tables \(Db2 Administration Guide\)](#).

Allow applications to specify a deadlock resolution priority

Function level 501 introduces the SYSIBMADM.DEADLOCK_RESOLUTION_PRIORITY built-in global variable to allow an application to specify a priority to use when resolving a deadlock situation with other threads. When an application sets and uses this built-in global variable (by using a SET *assignment-statement* SQL statement), the Db2 subsystem uses that value as a relative weighting factor to resolve deadlock situations with other threads.

For more information, see [DEADLOCK_RESOLUTION_PRIORITY \(Db2 SQL\)](#).

You can also use Db2 profile tables to specify values for the new SYSIBMADM.DEADLOCK_RESOLUTION_PRIORITY built-in global variable for both remote and local applications. See [Setting built-in global variables by using profile tables \(Db2 Administration Guide\)](#).

Profile table enhancements for application environment settings

Db2 13 introduces the capability to use system profiles for local applications in certain situations. Previously, the initial values for special registers and system built-in global variables can be specified in the Db2 profile tables, but they are used only for initialization with distributed threads. The new Db2 profile table support for local applications requires Db2 to be started with the DDF subsystem parameter set to AUTO or COMMAND. See [DDF STARTUP OPTION field \(DDF subsystem parameter\) \(Db2 Installation and Migration\)](#).

You can use Db2 profiles tables for both local and remote applications in the following situations:

Db2 function level

New Db2 profile table support

Function level 500 or higher

- You can specify assignments to the new CURRENT LOCK TIMEOUT special register. For more information, see [CURRENT LOCK TIMEOUT special register \(Db2 SQL\)](#) and [Setting special registers by using profile tables \(Db2 Administration Guide\)](#).
- You can specify a new RELEASE_PACKAGE keyword with a COMMIT attribute to change the release bind option for a package. See [Overriding the RELEASE\(DEALLOCATE\) option for packages by using profile tables \(Db2 Performance\)](#).

Function level 501 or higher

- You can specify values for the new SYSIBMADM.DEADLOCK_RESOLUTION_PRIORITY built-in global variable. See [DEADLOCK_RESOLUTION_PRIORITY \(Db2 SQL\)](#) and [Setting built-in global variables by using profile tables \(Db2 Administration Guide\)](#).

Record dependencies and validity at the statement level for packages

Starting in function level 502, when a package is bound or rebound with the new DEPLEVEL(STATEMENT) option, package invalidation is handled at a statement level. If one or more statements in a package are invalidated, a value of 'S' is recorded in the VALID column of the SYSPACKAGE or SYSPACKCOPY catalog table to indicate that the package is partially invalidated. The VALID column of the SYSPACKSTMT or SYSPACKSTMTCOPY catalog tables indicates the specific statement that is invalidated. In function level 503 or lower, a partially invalidated package is handled in the same manner as an invalidated package on REBIND and in an autobind situation.

Tip: [FL 504](#) This enhancement primarily introduces infrastructure to support a new capability that becomes available after you activate function level 504 or higher in Db2 13. For more information see [Enabling autobind phase-in for packages invalidated at the statement level \(Db2 Application programming and SQL\)](#).

The new dependencies recorded are only for statement-level object dependencies. Only changes to an object invalidate a package at the statement level. Any changes in authorization privileges still invalidate a package at the package level. For more information, see [Changes that invalidate packages \(Db2 Application programming and SQL\)](#).

Note: At function level 502 or higher, a package that is bound with DEPLEVEL(STATEMENT) option can be marked with VALID='S' if it is invalidated by a DROP FUNCTION statement. However, after the PTF for APAR PH55497 is applied, Db2 always marks a package that is invalidated by a DROP FUNCTION statement with VALID='N', regardless of the DEPLEVEL bind option for the package.

Package dependencies continue to be stored in the SYSPACKDEP catalog table. If a package is bound with the DEPLEVEL(STATEMENT) option, a new catalog table, SYSPACKSTMTDEP, stores the dependencies between a package statement and the objects referenced by that statement. The SYSPACKSTMTDEP catalog table contains the statement-level dependencies for a current package and all of the package copies.

Incompatible change:

An existing application might be marked with a value of 'S' in the VALID column of the SYSPACKAGE or SYSPACKCOPY table to indicate that the package has individual statements marked as invalid. There are no behavioral differences between how an invalidated package (VALID = 'N') and a partially invalidated package (VALID = 'S') are handled. The VALID value is reset to 'Y' in the same manner. However, if there are jobs or queries that look explicitly for 'N' in this column, these jobs might need to be modified to account for the new value of 'S'.

For more information, see [Chapter 4, “Incompatible changes in Db2 13,” on page 89](#).

For more information, see the following related topics:

- [DEPLEVEL bind option \(Db2 Commands\)](#)

- [PACKAGE DEPENDENCY LEVEL field \(PACKAGE_DEPENDENCY_LEVEL subsystem parameter\) \(Db2 Installation and Migration\)](#)
- [DSNTIP4: Application programming defaults panel 2 \(Db2 Installation and Migration\)](#)
- [SYSPACKSTMT catalog table \(Db2 SQL\)](#)
- [SYSPACKSTMTDEP catalog table \(Db2 SQL\)](#)
- [Changes that invalidate packages \(Db2 Application programming and SQL\)](#)

PH47560 delivered the functional code to record dependencies and validity for a package at the statement level.

ORDER BY support for a *fullselect* that invokes LISTAGG

Function level 504 removes a restriction against specifying an ORDER BY clause in a fullselect that contains an invocation of the LISTAGG built-in function for applications that run at application compatibility level V13R1M504 or higher. For more information, see [LISTAGG \(Db2 SQL\)](#).

APAR PH55596 delivered the functional code for ORDER BY support for a *fullselect* that invokes LISTAGG.

Minimized impact of invalidated packages with statement-level invalidation

Starting in function level 504, you can enable packages for statement-level invalidation. This new capability uses the statement-level dependency infrastructure, including the new DEPLEVEL(STATEMENT) bind option, that was introduced in function level 502. See [“Record dependencies and validity at the statement level for packages” on page 65](#).

With statement-level dependencies and statement-level invalidation, applications can execute a package that is invalidated at a statement level without waiting for the completion of the automatic rebind (autobind). The application thread can execute the valid statements immediately. Any invalid statements go through incremental bind in the application thread before execution. Concurrently, Db2 initiates a special autobind, which is called *autobind phase-in*. When the autobind phase-in finishes successfully, the subsequent executing threads can use the new valid copy of the package, and they no longer use incremental bind for any statements.

The autobind phase-in process runs concurrently with executing application threads, and the old invalid copy becomes a phased-out copy, which is stored in the SYSIBM.SYSPACKCOPY table. The new copy of the package becomes the current copy. Subsequent executions of the package use the new valid copy. Threads that existed prior to autobind phase-in completing can also use the new current copy when they release the phased-out copy (based on the RELEASE(COMMIT) or RELEASE(DEALLOCATE) bind options). In this respect, autobind phase-in is the same as rebind phase-in. However, unlike other phase-in rebinds, autobind phase-in does not require the PLANMGMT subsystem parameter to be set to EXTENDED. For more about rebind-phase in, see [Phase-in of package rebinds \(Db2 Application programming and SQL\)](#).

Db2 uses automatic binds only when the ABIND subsystem parameter is set to YES or COEXIST. If ABIND is set to NO when an invalid package runs, Db2 returns an error. For details, see [AUTO BIND field \(ABIND subsystem parameter\) \(Db2 Installation and Migration\)](#).

Incompatible change: If incremental binds fail due to prior invalidating changes to dependent objects, applications that previously failed with SQL code -904 might start receiving different SQL codes. For example, an incremental bind for a statement that depended on a dropped table might return -204 to the application. For more information, see [Chapter 4, “Incompatible changes in Db2 13,” on page 89](#).

Note: At function level 502 or higher, a package that is bound with DEPLEVEL(STATEMENT) option can be marked with VALID='S' if it is invalidated by a DROP FUNCTION statement. However, after the PTF for APAR PH55497 is applied, Db2 always marks a package that is invalidated by a DROP FUNCTION statement with VALID='N'.

For more information, see the following related topics:

- [Autobind phase-in for packages with statement-level invalidation \(Db2 Application programming and SQL\)](#)

- [Enabling autobind phase-in for packages invalidated at the statement level \(Db2 Application programming and SQL\)](#)
- [MAX_CONCURRENT_PKG_OPS in macro DSN6SPRM \(Db2 Installation and Migration\)](#)
- [SYSPACKAGE catalog table \(Db2 SQL\)](#)
- [00E30305 \(Db2 Codes\)](#)

APAR PH55497 delivered the functional code for minimizing the impact of invalidated packages.

SQL Data Insights

Function level 500 delivers SQL Data Insights (SQL DI), an integrated solution that brings deep learning AI capabilities into Db2. SQL DI uses unsupervised neural networks to generate a specialized vector-embedding model called database embedding, which can be referenced through SQL queries called "cognitive intelligence" queries.

The SQL DI user interface is an optional feature available at no additional charge with Db2 13, which provides the user interface for training models and exploring data insights. Db2 provides the in-database infrastructure for training and model table (vector table) management. Db2 also provides three new built-in cognitive functions to speed up query execution.

For more information, see [Running AI queries with SQL SQL Data Insights](#).

Enhancements for SQL Data Insights

Function level 504 introduces the following enhancements for the SQL DI capability in Db2 13:

- New Db2 built-in AI_COMMONALITY function that computes a similarity score by using the value of the *expression* argument and the centroid value of the model column. You can use the function in your queries to detect the common patterns and the outliers in your data.
- Enhanced AI_ANALOGY function with the numeric data type support. You can specify numeric data types in the arguments for the function.
- Automatic disablement of SQL DI vector prefetching based on the AI object type and the AI cache size. When the MXAIDTCACH parameter is set to a value greater than 0 and a query invokes a SQL DI function on a table, Db2 dynamically chooses between vector prefetching and row-by-row processing to optimize the CPU usage of the function.

APAR PH55212 delivered the functional code for these SQL DI enhancements.

Increased flexibility for package ownership

Starting at function level 500, you can specify the type of owner for a plan, package, or service, or the type of package owner for an SQL PL routine. The owner can be a role or an authorization ID. The default owner is a role in a trusted context that is defined with the role as object owner and qualifier attributes, otherwise the default owner is an authorization ID.

For more information, see the PACKAGE OWNER clause of CREATE PROCEDURE (SQL - native) (Db2 SQL) and the OWNERTYPE option of the OWNER bind option (Db2 Commands).

Column names longer than 30 bytes

Function level 100 extends the maximum length of a column name from 30 bytes of EBCDIC, up to 128 bytes with limited support for using the longer column names. The longer column names can be used when the TABLE_COL_NAME_EXPANSION subsystem parameter setting is ON. Although you can now define a column with a name up to 128 bytes, column names with a length greater than 30 bytes of EBCDIC might be truncated on a character boundary. Column names returned in an SQLDA contain 30 bytes at most. APIs that do not use the SQLDA to obtain a column name might return complete column names.

For more information, see [Column names longer than 30 bytes \(Db2 SQL\)](#) and [TABLE_COL_NAME_EXPANSION in macro DSN6SPRM \(Db2 Installation and Migration\)](#).

Memory usage reduced for REBIND with APREUSE

Function level 100 introduces reduced storage usage during BIND/REBIND for queries that involve many tables. This enhancement optimizes Db2 13 and improves its performance without changing how you configure, monitor, and use Db2.

Support for UNI_90 locale for the LOWER, TRANSLATE, and UPPER built-in functions

APAR PH47187 (September 2022) adds support to specify that the LOWER, TRANSLATE, and UPPER built-in functions use z/OS Unicode Conversion Services only with "normal" casing capabilities and use Unicode Standard 9.0.0. Db2 now supports a new locale, UNI_90, which specifies that z/OS Unicode Conversion Services is used with only "normal" casing capabilities and use Unicode Standard 9.0.0.

Before to this APAR, Db2 supports the following locales to indicate that Db2 should use z/OS Unicode Conversion Services to perform the upper or lower conversion with "normal" or "normal" and "special" casing capabilities.

- 'UNI' uses Unicode version 3.0.0 (which is the default for z/OS Unicode Services).
- 'UNI_SIMPLE' uses the latest version of the Unicode Standard that z/OS Unicode Conversion Services supports.
- 'UNI_60' specifies that z/OS Unicode Conversion Services is used with only "normal" casing capabilities and use Unicode Standard 6.0.0.

FL 504 Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [LOWER \(Db2 SQL\)](#)
- [UPPER \(Db2 SQL\)](#)
- [TRANSLATE \(Db2 SQL\)](#)
- [PH47187](#)

List prefetch for MERGE statements

APAR PH47581 (September 2022) introduces list prefetch as a possible access path for the MERGE statement. Index access with list prefetch can sometimes be used when an index column is being updated by the MERGE statement. Prior to this APAR, such MERGE statements would likely resort to table space scans. For more information, see the following related topics:

FL 504 Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

- [Index access for MERGE \(Db2 Performance\)](#)
- [MERGE \(Db2 SQL\)](#)
- [List prefetch \(PREFETCH='L' or 'U'\) \(Db2 Performance\)](#)
- [PH47581](#)

View management authorization enhancements

APARs PH54863 and PH54936 (July 2023) enhance Db2 13 at function level 100 or higher to allow database administrators (DBAs) with DBADM authority on databases to drop a view created for another user and select from a view without always requiring system level authority. APAR PH54936 supports the view management authorization enhancements when using RACF access control authorization exit for access control. With these APARs, DROP statements are allowed for a view if the privilege set includes DBADM authority on the database that contains one of the base tables and the subsystem parameter DBACRVW is set to YES. SELECT statements for a view are allowed if the privilege set includes the required privileges such as SELECT or EXECUTE on all the base objects.

FL 504 Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [DROP \(Db2 SQL\)](#)
- [Authorization for queries \(Db2 SQL\)](#)

- [View privileges \(RACF Access Control Module Guide\)](#)
- [DBADM \(Managing Security\)](#)
- [PH54863](#)
- [PH54936](#)

New default values in existing rows for added ROW CHANGE TIMESTAMP columns

Starting at application compatibility level V13R1M503 or higher, Db2 uses a constant default value when a new ROW CHANGE TIMESTAMP column is added. When processing ALTER TABLE statements that specify ADD COLUMN for ROW CHANGE TIMESTAMP columns, Db2 now sets the corresponding value in the DEFAULTVALUE column value in the SYSIBM.SYSCOLUMNS catalog table to the timestamp of the ALTER TABLE statement. (Note that CREATE TABLE processing to define a ROW CHANGE TIMESTAMP column does not set the DEFAULTVALUE column value.)

Before this change, Db2 derives the default values for existing rows from the page header from the row, which is the RBA for standalone Db2 subsystems. In data sharing, the default is derived is based on an internal mapping table between the LRSN and a timestamp. As a result, inserts, deletes, or updates to any rows in a page can change the derived default row change timestamp column values for unchanged rows, leading to unpredictable results.

For more information, see the following related topics:

- [ROW CHANGE expression \(Db2 SQL\)](#)
- [SYSCOLUMNS catalog table \(Db2 SQL\)](#)
- [ALTER TABLE \(Db2 SQL\)](#)

Starting at function level 503 or higher, redirected recovery processing is also updated to allow different ROW CHANGE TIMESTAMP column default values between source and target tables. For more information, see [Running a redirected recovery \(Db2 Utilities\)](#).

APAR PH51185 delivered the functional code for improve default behavior for added ROW CHANGE TIMESTAMP columns.

Accelerator-only support for more than 32K elements in an IN list

Starting in function level 503, a query that has an IN list of more than 32,767 (32K) elements in an IN predicate can run as an accelerator-only query on IBM Db2 Analytics Accelerator V7 for z/OS, if all of the following conditions are met:

- Query acceleration is enabled and requested for the query. For more information, see [Enabling acceleration of SQL queries \(Db2 Performance\)](#).
- The application runs at Db2 application compatibility level V13R1M503 or higher.
- Option 12 is specified in the list of values for the [QUERY_ACCEL_OPTIONS](#) subsystem parameter.
- The target accelerator is IBM Db2 Analytics Accelerator V7 for z/OS.
- The IN list specifies only SQL constants.
- The query can be functionally supported by the target accelerator and IBM Db2 Warehouse.

When such queries run on a V7 accelerator, the enforcement of any limits on the number of elements in the IN list predicate is handled by IBM Db2 Warehouse, instead of Db2 for z/OS. IBM Db2 Warehouse has no documented limit for elements in an IN list. However, practical limitations such as the memory and processing resources available for the query in IBM Db2 Warehouse are still likely to impose some limit on the number of elements that can be specified in the IN list predicate.

A view can also be created in Db2 for z/OS for a query with an IN list that contains more than 32K elements, but only under the same conditions listed previously for queries that contain such IN predicates. Any query that uses such a view is also evaluated and limited to the same conditions and restrictions described above. Such views also have the value 'R' in IBMREQD column in the SYSIBM.SYSVIEWS catalog table, to indicate the Db2 13 release dependency.

For more information, see "Accelerator-only support for more than 32K elements in an IN list" in [IN predicate \(Db2 SQL\)](#).

APAR PH50756 delivered the functional code for the accelerator support for large IN list predicates.

Freeing inactive packages for native REST services

Starting in Db2 13 with APAR PH54129 (June 2023), you can issue the DSN FREE PACKAGE command to free inactive packages for native REST services.

Db2 native REST services are created by invoking the REST service manager API with a `createService` request, or by using the DSN BIND SERVICE command. This inserts a row into the REST services table defining the service and creates an application package. You can then rebind the application package for a REST service with various options by using the DSN REBIND PACKAGE command. During rebind, inactive package copies, such as the original and previous copies, can be generated based on the PLANMGMT option. Phased-out copies of packages can be generated if already running threads have the package allocated when the rebind occurs. Invalid package copies can occur when a dependent object is changed or dropped. These types of inactive packages can build up over time when many rebinds are done.

Before this APAR, the only method of freeing inactive REST service packages is by invoking the REST service manager API with a `dropService` request, or by use of the FREE SERVICE command. The shortcoming to doing this is that the REST service itself is freed and all packages, both active and inactive, are freed. Also the FREE PACKAGE command allows the freeing of inactive packages only if they are not REST services, and is blocked for REST services with error message DSNT246I.

Use of the FREE PACKAGE command to free an active REST service package continues to be blocked with the DSNT264I message.

FL 504 Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [Freeing inactive packages for Db2 REST service \(Db2 REST services\)](#)
- [FREE PACKAGE \(DSN\) \(Db2 Commands\)](#)
- [FREE SERVICE \(DSN\) \(Db2 Commands\)](#)
- [DSNT264I \(Db2 Messages\)](#)
- [PH54129](#)

BIND QUERY restriction removed for certain SQL PL packages

APAR PH54056 (October 2023) relaxes a restriction against a BIND QUERY command that is issued to enforce statement-level plan hints for certain SQL PL packages.

Before this APAR, Db2 issues message DSNT281I with reason code 5 when processing a statement that has a DSN_USER_QUERY_TABLE row with a TYPE='N' column value. This restriction prevents the enforcement of statement-level access path hints for statements in SQL PL packages, which have the TYPE='N' value in the SYSPACKAGE catalog table.

For more information, see the following related topics:

- [Specifying access paths at the statement level \(Db2 Performance\)](#)
- [BIND QUERY \(DSN\) \(Db2 Commands\)](#)
- [PH54056](#)

IBM zSystems hardware synergy in Db2 13

Db2 13 introduces the following new capabilities that take advantage of the synergy between IBM zSystems hardware and Db2 for z/OS.

Expanded SORTL usage with learning from execution (IBM z15®)

Function level 100 introduces expanded SORTL usage based on machine learning on the amount of storage and the number of records being sorted, when run on IBM z15 or later processors. This enhancement optimizes Db2 13 and improves its performance without changing how you configure, monitor, and use Db2.

Db2 support for z/OS continuous compliance

Customers are looking for solutions that provide evidence that they can trust the security of z/OS systems. z/OS 2.5 introduces new SMF type 1154 records that provide evidence of security compliance. Participating components and products can collect and write compliance data to their associated SMF 1154 subtype records. Function level 100 adds the capability to collect evidence on Db2 subsystems' compliance by writing SMF 1154 subtype 81 records. For more information, see [Db2 evidence for z/OS continuous compliance \(Managing Security\)](#) and [What is new in z/OS \(V2R4 - V2R5\)](#).

IBM z16™ group buffer pool (GBP) residency time

Starting in function level 100, two new statistics are added to relevant group buffer pool statistics storage areas:

- The weighted-average time a data area resides in a storage class before it is reclaimed, or 0 if the group buffer pool has not been reclaimed.
- The weighted-time a directory entry resides in a storage class before it is reclaimed, or 0 if the group buffer pool has not been reclaimed.

The new residency time statistics are supported only if Db2 13 runs in the following environment:

- z/OS 2.4 or later, with PTFs required for *cache residency time metrics* applied.
- IBM z16 or later processors.
- Coupling facility at CF Level 25 or higher. For more information, see [CFLEVEL and operating system level coexistence \(z/OS MVS Setting Up a Sysplex\)](#).

You can access these metrics with the IFCID record trace and the -DISPLAY GROUPBUFFERPOOL command. For more information, see "DSNB820I: Average residency times" in [DSNB750I \(Db2 Messages\)](#).

DECFLOAT datatype support for PL/I and C with DCLGEN

APAR PH47453 (August 2022) enhances DCLGEN to support the DECFLOAT data type for the PL/I and C programming languages.

[FL 504](#) Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [Data types that DCLGEN uses for variable declarations \(Db2 Application programming and SQL\)](#)
- [DCLGEN \(declarations generator\) \(Db2 Application programming and SQL\)](#)
- [PH47453](#)

Availability in Db2 13

Db2 13 introduces the following new capabilities that improve the online availability of your data and database applications on Db2 for z/OS.

DBAT availability improvements

Function level 100 introduces changes to Db2 13 DBAT termination processing to support the following objectives:

- Reduction of the overall frequency and number of DBAT terminations.
- Reduction of the number of concurrent DBAT terminations that are caused by a short-term increase in DBAT usage.

This enhancement optimizes Db2 13 and improves its performance without changing how you configure, monitor, and use Db2.

Online conversion of tables from growth-based (PBG) to range-based (PBR) partitions

Function level 500 introduces the capability to convert the partitioning scheme of a table with growth-based partitions (in a PBG table space) to use range-based partitions (in a PBR table space). The conversion can be completed as an online change with minimal impact to your applications.

PBG and PBR universal table spaces (UTS) are the strategic table space types for tables in Db2 for z/OS. PBG table spaces are the default UTS type, and they are well-suited for small to medium-sized tables. However, if an existing table in a PBG table space grows too large, performance degradation or data and index management issues might arise. Consider converting from PBG to PBR when that occurs.

To complete the conversion, you issue an ALTER TABLE statement with the new ALTER PARTITIONING TO PARTITION BY RANGE clause and run the REORG TABLESPACE utility to materialize the pending change. The table space for the table is converted to PBR with relative page numbering (RPN).

For more information, see [Converting tables from growth-based to range-based partitions \(Db2 Administration Guide\)](#) and "ALTER PARTITIONING TO PARTITION BY RANGE" in [ALTER TABLE \(Db2 SQL\)](#).

Removed stacking limitations for PBG to PBR conversions

Starting in function level 500 or higher with APAR PH51359 (December 2022), Db2 13 supports *stacking* of certain pending data definition changes when a table in a partition-by-growth (PBG) table space is converted to partition-by-range (PBR). That is, the pending definition changes in following table can now be issued together and materialized by a single execution of the REORG utility. This capability is especially useful if you need to enlarge the partition data set sizes to accommodate the distribution of data into the partitions, alter the columns to be used as partitioning keys, or alter other table space or index attributes.

Object level	Supported stacked pending definition changes for PBG to PBR conversion
Table space	<ul style="list-style-type: none"> • BUFFERPOOL • DSSIZE • SEGSIZE (excluding conversion to UTS) • MEMBER CLUSTER
Table	<ul style="list-style-type: none"> • ALTER COLUMN • DROP COLUMN
Index	<ul style="list-style-type: none"> • BUFFERPOOL • COMPRESS

Before this change, Db2 issues SQLCODE -20385 if you try to issue any of these alterations when a PBG to PBR conversion is pending, or in the opposite situation, so at least two executions of the REORG utility are required to complete any of these changes if they are needed for the conversion.

FL 504 Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [Converting tables from growth-based to range-based partitions \(Db2 Administration Guide\)](#)
- [Restrictions for pending data definition changes \(Db2 Administration Guide\)](#)
- [-20385 \(Db2 Codes\)](#)
- [PH51359](#)

Ability to delete an active log data set from the BSDS while Db2 is running

Function level 500 introduces the new REMOVELOG option for the -SET LOG command to support online removal of an active log data set from the BSDS, eliminating the need to stop Db2 to accomplish the task by using the offline utility DSNJU003. The -SET LOG REMOVELOG command deletes the specified log from the BSDS if it is not in use or mark the log REMOVAL PENDING if it is in use.

To provide monitoring of the current active log status for log data sets with REMOVAL PENDING status, function level 500 also introduces the DETAIL option for the -DISPLAY LOG command. It shows information regarding REMOVAL PENDING status for local active log data sets. The output from the utility DSNJU004 also shows the REMOVAL PENDING status where applicable.

For more information, see [Deleting an active log data set from the BSDS with the -SET LOG command \(Db2 Administration Guide\)](#).

Relative page numbering for new PBR table spaces

Starting in function level 100, the default value of the PAGESET_PAGENUM subsystem parameter is changed to RELATIVE. The PAGESET_PAGENUM subsystem parameter specifies the default value that Db2 uses when you omit the PAGENUM option in CREATE TABLESPACE or CREATE TABLE statement for a partition-by-range (PBR) table space. That is, it specifies whether Db2 creates the table space and associated partitioned indexes to use relative page numbers (RPN) or absolute page numbers (APN) across partitions. RPN is the strategic direction for PBR table spaces in Db2. If you accept the new default and create all new PBR table spaces with relative page numbers, you can avoid costly future conversions. Converting from absolute to relative page numbers always requires a REORG of the entire table space.

See [PAGE SET PAGE NUMBERING field \(PAGESET_PAGENUM subsystem parameter\) \(Db2 Installation and Migration\)](#).

Improved concurrency for altering tables for DATA CAPTURE

Function level 500 introduces a concurrency improvement for ALTER TABLE statements that change the DATA CAPTURE attribute of tables. With this enhancement, Db2 no longer waits for other statements that depend on the altered table to commit. As a result, the DATA CAPTURE alteration can now succeed even when concurrent statements are running continually against the table.

Earlier Db2 releases quiesce the following objects that depend on the altered table as part of the DATA CAPTURE alteration:

- Static packages
- Cached dynamic SQL statements

Because the DATA CAPTURE alteration waited for applications that depended on the altered table to commit, continuous concurrent activity on the table might cause the ALTER TABLE statements to fail.

The new DATA CAPTURE attribute now takes effect immediately when the processing completes, even before the ALTER statement commits. As a result, concurrent statements on the same Db2 member might write out different log formats in the same transaction. For more information, see [Altering a table to capture changed data \(Db2 Administration Guide\)](#).

CREATE TABLESPACE uses MAXPARTITIONS 254 by default

At application compatibility level V13R1M500 or higher, CREATE TABLESPACE statements use MAXPARTITIONS 254 by default.

When MAXPARTITIONS 256 is explicitly specified, the default DSSIZE varies from 4 G to 32 G depending on the page size. However, starting with application compatibility level V12R1M504, when MAXPARTITIONS is not explicitly specified, Db2 12 uses MAXPARTITIONS 256 by default, but the default DSSIZE is always 4 G regardless of the page size.

This apparent inconsistency avoided a risk of failure for existing statements, where the default data set size might be greater than 4 G depending on the page size. The statements might fail with SQLCODE -904 with reason code 00D70008 if the data sets for the table space are not associated with a DFSMS data class that is specified with extended format and extended addressability.

With MAXPARTITIONS 254 as the default, the result is now consistent regardless of whether MAXPARTITIONS is explicitly specified. The calculated default DSSIZE is always 4 G.

See the MAXPARTITIONS and DSSIZE descriptions in [CREATE TABLESPACE \(Db2 SQL\)](#).

Db2 controlled sysplex workload balancing

Starting in function level 502, you can enable sysplex workload balancing (WLB) on a Db2 for z/OS server.

sysplex WLB functionality (also called *transaction-level workload balancing*) on Db2 for z/OS servers provides high availability for client applications that connect directly to a data sharing group by balancing work among members of a data sharing group at the start of a transaction. Db2 for z/OS server returns a list of available members with weights provided by WLM . At the start of a transaction, the client application chooses the member with the best weight from the list. Configuration parameters are available on clients (Java and non-Java) to enable this functionality.

To enable use of this new capability, function level 502 introduces the following new keywords for the `-MODIFY DDF (Db2) (Db2 Commands)` command.

RQSTWLB

Specifies that Db2 returns to the remote client a request to enable sysplex workload balancing. This option can only be specified if the subsystem is part of a data sharing group. This option has group-wide scope. The decision to honor sysplex workload balancing request lies with the client.

Notes:

- The RQSTWLB option applies for Db2 for Linux[®], UNIX, and Windows 11.5 or later (CLI Db2 code release "SQL11050" and IBM Data Server Driver for JDBC and SQLJ 4.26.14 or later.)
- If RQSTWLB is used, applications might encounter additional communication errors such as SQL codes -30108 and -20542. These errors are removed by default with Db2 11.5.8 or later client drivers, and the client property `enableseamlessfailovererrorcodes` controls whether they are issued.

★

The option is enabled for entire data-sharing group. This is the default option.

location-name

The option is enabled for remote clients that access the data-sharing group using the location-name as the data source.

alias-name

The option is enabled for remote clients that access the data-sharing group using the specified alias-name as data source.

DFLTWLB

Db2 honors the sysplex workload balancing option requested by the client. This option can only be specified if the subsystem is part of a data sharing group. This option has group-wide scope. DFLTWLB is the default installed option.

★

The option is enabled for entire data-sharing group. This is the default option.

location-name

The option is enabled for remote clients that access the data-sharing group using the location-name as the data source.

alias-name

The option is enabled for remote clients that access the data-sharing group using the specified alias-name as data source.

For more information about enabling sysplex workload balancing with client drivers, see:

- [Java client support for high availability on IBM data servers](#)
- [Non-Java client support for high availability for connections to Db2 for z/OS servers](#)

PH48253 delivered the functional code for Db2 controlled sysplex workload balancing.

Scalability in Db2 13

Db2 13 introduces the following capabilities to improve the scalability of your data and database applications on Db2 for z/OS.

Reduced ECSA storage for IFI buffers

Db2 13 reduces the use of ECSA storage for IFI buffers from a maximum of 50 MB to a fixed 8 MB.

Function level 100 reduces the use of ECSA storage for IFI buffers to a maximum of 25 MB. Then, after function level 500 is first activated, it is further reduced to 8 MB. The storage behavior that is introduced in function level 500 continues even if you later activate function level 100*.

To compensate for the reduction in ECSA storage, you must set aside an extra 50 MB for HVCOMMON and 25 MB for private storage. You can reduce the ECSA storage after function level 500 is activated and Db2 starts using the new storage pools. When Db2 uses the new storage pools, the use of ECSA for the retrieval of IFI records noticeably decreases. You can monitor use of the new storage pools by starting the statistics trace to collect IFCID 225. Then, you can check the SHARED / COMMON storage summary report in the formatted IDCID 225 SMF trace record.

For more information about ECSA storage requirements, see [Calculating the storage requirement for the extended common service area \(Db2 Installation and Migration\)](#).

Reduced ECSA storage use for distributed data facility (DDF) processing

Function level 100 reduces the amount of ECSA storage that is used for processing DDF threads to be equivalent to processing local threads. The previous recommendation was an extra 2 KB per DDF thread. For more information, see [Calculating the storage requirement for the extended common service area \(Db2 Installation and Migration\)](#).

Reduced agent local below-the-bar (BTB) storage

Starting in function level 100, Db2 supports a greater number of concurrent threads, by using above-the-bar (ATB) agent-local storage for statement text and attribute strings for dynamic SQL statements. In earlier releases, Db2 kept a copy of dynamic SQL statement text and attribute strings in agent local below-the-bar (BTB) storage while the statement is being prepared and executed.

For any specific thread, multiple dynamic SQL statements can be executing depending on the nesting level. The maximum length of an SQL statement is 2 MB, but much more storage can be allocated and the consumption of BTB storage could prevent the number of threads from scaling.

This enhancement optimizes Db2 13 and improves its performance without changing how you configure, monitor, and use Db2.

Improved storage monitoring and contraction

Function level 100 introduces the following enhancements to provide storage constraint relief:

- When below-the-bar Db2 storage consumption exceeds 64-percent threshold, Db2 automatically begins contraction of private storage pools.
- When extended common service area (ECSA) storage consumption exceeds the 85-percent threshold, Db2 automatically begins contraction of storage pools that are allocated in the ECSA.

In both cases, the storage contraction stops after storage consumption drops below the threshold.

This enhancement optimizes Db2 13 and improves its performance without changing how you configure, monitor, and use Db2.

Dynamic management of CF lock storage by IRLM

With IRLM 2.3 at function level 50C or higher, which is included with Db2 13, IRLM can now invoke an existing capability in z/OS Sysplex Services for Data sharing (XES) to dynamically expand the coupling facility (CF) lock structure storage size. This new internal monitoring capability in IRLM can improve lock request processing throughput, by expanding the CF lock structure size to process lock requests, instead of rejecting them.

The existing XES monitoring of the CF lock structure use is defined in the coupling facility resource management (CFRM) policy as a threshold percentage value, and it is enabled with a default of 80%

when it is not equal to zero. This monitoring retrieves statistics on the CF lock structure every 60 seconds when the storage usage is less than the threshold and every 30 seconds when the storage usage is equal to or greater than the full threshold. IRLM can determine the storage needed at a higher level of granularity than the existing monitoring by XES of CF structure, especially when a spike in locking activities results in rejection of lock requests due to insufficient Record List Entries (RLEs) even before z/OS has a chance to start the CF lock structure alteration.

The existing structure monitoring by XES handles storage contraction, and it contracts all eligible structures in the coupling facility by 10 percent in each cycle when the entire coupling facility is at or more than 90% full.

IRLM issues the following messages when it adjusts the CF lock structure storage size: [DXR189I](#) and [DXR190I](#).

Improved Db2 installation and migration process for customizing the amount of private storage for IRLM locks

In the MAX STORAGE FOR LOCKS field on installation panel DSNTIPJ, you can specify the maximum amount of private storage above the 2 GB bar for the IRLM lock control structure. In earlier Db2 releases, you can specify a value of only up to 102400 megabytes. Starting in Db2 13, you can specify a value of up to 16384 petabytes.

For more information, see [MAX STORAGE FOR LOCKS field \(Db2 Installation and Migration\)](#) and [MAX LOCK STORAGE UNIT field \(Db2 Installation and Migration\)](#).

Real-time statistics scalability

As data volumes become larger, the widths of some columns in the real-time statistics tables are not large enough to accommodate larger values. In addition, during high volume processing, lock escalation might occur on the real-time statistics history table spaces. Lock escalation can negatively affect concurrency and performance.

Starting after function level 501 is activated, the following changes to the real-time statistics tables and table spaces are introduced to provide greater capacity and concurrency:

- In real-time statistics tables SYSIBM.SYSTABLESPACESTATS and SYSIBM.SYSINDEXSPACESTATS, and their associated history tables SYSIBM.SYSTABSPACESTATS_H and SYSIBM.SYSIXSPACESTATS_H, some column data types are BIGINT instead of INTEGER, or INTEGER instead of SMALLINT.
- Lock escalation is disabled on the following table spaces: DSND06.SYSTSTSS and DSND06.SYSTSISS for the RTS tables; and DSND06.SYSTSTSH and DSND06.SYSTSISH for the RTS history tables

For more information, see [SYSTABLESPACESTATS catalog table \(Db2 SQL\)](#) and [SYSINDEXSPACESTATS catalog table \(Db2 SQL\)](#).

This enhancement optimizes Db2 13 and improves its performance without changing how you configure, monitor, and use Db2.

More efficient cleanup for above-the-bar storage

Function level 100 introduces improvements to how Db2 manages and frees above-the-bar storage, especially to reduce the disruptive impact of issuing excessive IARV64 REQUEST(DISCARDDATA) service requests.

Db2 13 no longer issues the IARV64 REQUEST(DISCARDDATA) request during thread deallocation or at certain intervals of COMMIT, and enhanced storage management is no longer controlled by the REALSTORAGE_MANAGEMENT subsystem parameter, which is also removed. In Db2 13, the storage is returned to the memory object. A system-level timer drives contraction for the memory object to release unused frames back to z/OS. Also, Db2 13 periodically checks the available free frames before the LPAR starts to page (by using the z/OS calculations for available free frames and LO threshold). If this value becomes lower than 5 times the z/OS calculated OK threshold, the memory object contraction is triggered.

SPT01 and SYSLGRNX table spaces are converted to DSSIZE 256 GB

Starting in function level 500, the first time that the REORG TABLESPACE utility runs for the following directory objects, it converts the DSSIZE to 256 GB.

- DSNCDB01.SPT01 to resolve issues that are related to the removal of the SPT01_INLINE_LENGTH subsystem parameter by APAR PH24358 in Db2 12.
- DSNCDB01.SYSLGRNX in anticipation of future growth in this table for increasing workloads and conversions of non-UTS table space to UTS.

The conversion is automatic and does not require any special utility syntax. For more information, see [“Function level 500 \(for migrating to Db2 13 - May 2022\)” on page 71.](#)

More concurrent open data sets in z/OS 2.5

In function level 100 and z/OS 2.5 or later, dynamic allocation processing supports system work blocks (SWBs) for data sets that are in 64-bit storage. This new dynamic allocation function helps reduce below-the-bar storage usage for address spaces that allocate large numbers of data sets.

To enable this feature, complete one of the following actions:

- Update the ALLOCxx parmlib member to set the SYSTEM SWBSTORAGE value to ATB. The default value is SWA, which indicates that SWBs reside in 24-bit storage or 31-bit storage. ATB indicates that SWBs are allowed to reside in 64-bit storage.
- Issue system command SETALLOC SYSTEM,SWBSTORAGE=ATB.

It is best to update the ALLOCxx parmlib member because the change remains effective across IPLs. If the SETALLOC command is used to enable SYSTEM SWBSTORAGE, you must restart Db2 for the change to take effect.

Previously, the CLIST calculation for data set storage size used 5 KB per open data set. With the new dynamic allocation function, the storage that is required per open data set is reduced to 4 KB. You must adjust the calculation for data set storage size. For more information, see [Calculating data set control block storage \(Db2 Installation and Migration\).](#)

Performance in Db2 13

Sort optimization

Function level 100 introduces enhanced sort optimizations, which were previously introduced for ORDER BY and GROUP BY processing. It applies them to improve the performance of certain operations, such as the following processing enhancements:

- Machine-generated code support for DECFLOAT processing.
- Support for the following enhancements for GROUPING SET, multiple DISTINCT, and PERCENTILE processing:
 - Machine-generated code support.
 - Sort processing can use its own work file.
 - A check for ordered data in the first iteration of a sort.
 - Larger sort trees can be used.
- SUBSTR support for the LISTAGG built-in function, if the start position and length for SUBSTR is a constant.
- Support for avoiding rereading of a single work file, if the sort work file is used, and if IBM Watson[®] Machine Learning for z/OS is enabled.
- Support for reducing the length of long VARCHAR keys, if the last key in an ORDER BY list is a VARCHAR over 100 bytes, and if IBM Watson Machine Learning for z/OS is enabled.

This enhancement optimizes Db2 13 and improves its performance without changing how you configure, monitor, and use Db2.

Improved locking for INSERT to partition-by-growth (PBG) table spaces

Function level 100 introduces retry logic for INSERT operations. An extra attempt is made to obtain a partition lock on a PBG table space after a failed first attempt, thereby increasing the success rate of INSERT operations.

Before this enhancement, only a single attempt was made to obtain a lock on the target partition. If the attempt failed, the target partition was skipped, and the next partition was evaluated. This process would continue until the INSERT operation either successfully obtained a partition lock or it finished searching all existing partitions without obtaining a partition lock.

In most cases, the duration of partition lock contention is short; however, because the INSERT operation did not make another attempt to obtain a lock on a partition after the first failed attempt, the INSERT operation terminated unnecessarily. In many cases, making an extra attempt to obtain a partition lock results in the successful completion of an INSERT operation that otherwise would fail.

This enhancement optimizes Db2 13 and improves its performance without changing how you configure, monitor, and use Db2.

Improved cross-partition search for INSERT to PBG table spaces

Starting in function level 100, if free space is not found in the initial target partition for an insert to a partition-by-growth (PBG) table space, Db2 13 more efficiently uses trailing empty partitions other than the last physical partitions.

In Db2 12 and later, the cross-partition search for PBG table spaces is bidirectional. That is, when an insert operation fails to find free space in the initial target partition, which is selected based on the clustering index, the partition to be searched next can be either an ascending or descending partition sequence number. The searching order is randomly decided at run time to avoid creating a “hot spot” in a single partition. When Db2 12 reaches the first physical partition during a descending partition search, it wraps around and looks at the last physical partition next. As a result, when a PBG table space has many empty partitions at the end, the descending cross-partition search algorithm in Db2 12 often uses the last physical partitions, and it can sometimes leave many unused empty partitions unused in between the first and last partitions.

To prevent this situation, Db2 13 now tracks the highest non-empty partition in the table space in real-time statistics at run time. The tracking starts when a data set is opened and continues until it is closed. When Db2 13 uses a descending cross-partition search and after it reaches the first partition, Db2 13 now searches the cached highest non-empty partition next, instead of the last physical partitions.

This enhancement optimizes Db2 13 and improves its performance without changing how you configure, monitor, and use Db2.

Index look-aside optimization

Function level 100 introduces index look-aside optimizations, to improve performance for insert, update, and delete operations. This enhancement optimizes Db2 13 and improves its performance without changing how you configure, monitor, and use Db2.

Improved performance when using external security

Function level 100 introduces the following enhancements to improve performance for Db2 environments that use external security:

- Db2 caches plan authorization checks that the access control authorization exit (ACAE) routine uses. Previously, successful authorization checks on the EXECUTE privilege for plans were not cached if those checks were completed by the ACAE routine. This enhancement provides consistent behavior in plan authorization cache behavior regardless of whether security is managed with Db2 facilities or with the ACAE.

To enable plan authorization caching when the ACAE routine is being used, the AUTHEXIT_CACHEREFRESH subsystem parameter must be set to ALL and the z/OS release must be 2.5 or later. Db2 caches the results of authorization checks on the EXECUTE privilege for plans if a profile in the RACF class MDSNPN permits access to the plan. Db2 does not cache the results if access is allowed due to administrative authority, such as the DATAACCESS or SYSADM authorities.

For more information, see [Caching of EXECUTE on plans, packages, and routines \(Managing Security\)](#).

- Db2 is enhanced to cache more authorization IDs per plan. For more information, see [Caching authorization IDs for plans \(Managing Security\)](#).
- The AUTHCACH subsystem parameter is removed to simplify plan authorization cache management. Use the CACHESIZE bind option on the BIND PLAN subcommand to specify the size of the authorization cache for that plan. The default value is 4K.
- If the AUTHEXIT_CACHEREFRESH subsystem parameter is set to ALL, the global authentication cache takes the timestamp into consideration for user IDs that were authenticated by using credentials other than multi-factor authentication (MFA). For more information, see [Global authentication cache \(Managing Security\)](#).
- When you specify a key label for data set encryption, the specified key label cannot refer to an archived key for decryption operations only. Key labels can be specified by using the ENCRYPTION_KEYLABEL subsystem parameter or any of the following SQL statements:
 - ALTER STOGROUP
 - ALTER TABLE
 - CREATE STOGROUP
 - CREATE TABLE

If the specified key label refers to a decryption-only archived key, the key label specification fails and returns an error message. For more information on decryption-only archived keys, see [ICSF: Limit archived keys to decrypt operations only](#).

Fast index traversal (FTB) support for larger index keys

Function level 500 extends FTB support to unique indexes with a key size for the ordering columns up to 128 bytes and nonunique indexes with a key size up to 120 bytes. For more information, see [Fast index traversal \(Db2 Performance\)](#).

Fast index traversal (FTB) controlling rows in SYSIBM.SYSINDEXCONTROL are read more frequently by Db2

PH57071 (October 2023) increases how frequently Db2 reads newly inserted rows for control of fast index traversal (FTB) processing in the SYSIBM.SYSINDEXCONTROL table. With this APAR, Db2 applies the newly inserted rows every 2 minutes, instead of every 10 minutes.

For more information, see the following related topics:

- [Enabling or disabling fast index traversal at the index level \(Db2 Performance\)](#)
- [SYSINDEXCONTROL catalog table \(Db2 SQL\)](#)
- [PH57071](#)

SELECT INTO statement support for OPTIMIZE FOR n ROWS

Starting at application compatibility level V13R1M503 or higher, you can specify an *optimize-clause* in SELECT INTO statements to enable Db2 to consider access paths that use a sort. SELECT INTO statements always return a single row. However, you can use OPTIMIZE FOR 2 ROWS can be specified to influence the Db2 optimizer.

When FETCH FIRST 1 ROW ONLY is specified, Db2 also applies a sort avoidance preference that is associated with OPTIMIZE FOR 1 ROWS during access path selection. However, sometimes the avoiding of any sort can result in a more expensive (in total cost) access path being used. If this situation occurs, OPTIMIZE FOR 2 ROWS can be specified in the SELECT INTO statement to enable Db2 to consider use of a more efficient access path that uses a sort. Other integer values can be specified for *n* in OPTIMIZE FOR *n* ROWS, but OPTIMIZE FOR 2 ROWS is recommended for this scenario.

For more information, see the following related topics:

- [SELECT INTO \(Db2 SQL\)](#)
- [optimize-clause \(Db2 SQL\)](#)

- [Interaction between FETCH and OPTIMIZE FOR clauses \(Db2 Performance\)](#)

APAR PH50010 delivered the functional code for *optimize-clause* support for SELECT INTO statements.

Db2 Utilities in Db2 13

Db2 13 introduces the following new capabilities and enhancements to Db2 Utilities Suite for z/OS:

Collection of real-time and historical information about utility execution

To improve utility management, function level 501 introduces the ability to collect real-time and historical information about utility execution. After you activate utility history collection by setting the UTILITY_HISTORY subsystem parameter to UTILITY, information about utilities is added to the SYSIBM.SYSUTILITIES catalog table. One row is inserted into the SYSUTILITIES table at the start of each utility execution. Then, information in the row is updated as the utility progresses, and final information is updated in the row when the utility execution finishes.

For more information, see [Monitoring utility history \(Db2 Utilities\)](#), [UTILITY HISTORY \(UTILITY_HISTORY subsystem parameter\) \(Db2 Installation and Migration\)](#), and [SYSUTILITIES catalog table \(Db2 SQL\)](#).

Utility object-level history

Starting in function level 504, you can collect object-level information along with utility execution information for IBM Db2 Utilities. Real time and historical information about the table spaces and index spaces for IBM Db2 Utilities executions are inserted to the SYSIBM.SYSOBJEVENTS catalog table.

System administrators and DBAs can use queries to generate, filter, order, and join the information in the SYSIBM.SYSOBJEVENTS and SYSIBM.SYSUTILITIES catalog tables to check, analyze, and compare utility executions.

For more information, see the following related topics:

- [Monitoring utility history \(Db2 Utilities\)](#)
- [Utility object-level history information in SYSOBJEVENTS \(Db2 Utilities\)](#)
- [UTILITY HISTORY \(UTILITY_HISTORY subsystem parameter\) \(Db2 Installation and Migration\)](#)
- [SYSUTILITIES catalog table \(Db2 SQL\)](#)
- [SYSOBJEVENTS catalog table \(Db2 SQL\)](#)

The following APARs delivered the functional code for utility object-level history: PH55476, PH55914, PH55915, and PH55916.

Page sampling for inline statistics

Beginning in function level 500, the REORG TABLESPACE and LOAD utilities can now use page sampling when they gather inline statistics. Page sampling has the potential to reduce both CPU time and elapsed time. In earlier Db2 releases, the RUNSTATS utility can use page sampling, but inline statistics that are gathered by other utilities can use row sampling only. To use page sampling for inline statistics with REORG TABLESPACE or LOAD, specify the TABLESAMPLE SYSTEM option or ensure that the STATPGSAMP subsystem parameter is set to SYSTEM (the default) or YES. In function level 500, STATPGSAMP is extended to apply to inline statistics. For more information, see the [TABLESAMPLE SYSTEM option description in Syntax and options of the LOAD control statement \(Db2 Utilities\)](#) and [Syntax and options of the REORG TABLESPACE control statement \(Db2 Utilities\)](#).

Enhanced space-level recovery with the RECOVER utility

Starting in function level 100, the RECOVER utility supports space-level recovery (where DSNUM ALL is specified or is the default), even if the image copies were created at the partition or piece level for the following objects:

- Universal table spaces (UTS).
- Index spaces or indexes for a UTS.

- XML UTS with a base table that resides in a UTS.
- Auxiliary index spaces or indexes for an XML UTS with a base table that resides in a UTS.
- LOB table spaces with a base table that resides in a UTS.
- Auxiliary index spaces or indexes for LOB table spaces with a base table that resides in a UTS.

Space-level recovery of these supported object types is processed as partition-level or piece-level recoveries in the RESTORE phase. Db2 catalog and directory objects that meet the criteria above are supported by this enhancement. When the list contains a mixture of supported and unsupported object types, recovery behavior for the unsupported object types is the same as Db2 12 or earlier releases.

When the RECOVER utility is invoked (with DSNUM ALL specified or as the default) in Db2 12 or earlier releases, for space-level recovery of table spaces, index spaces, or indexes, an error message is issued if the image copies were created at the partition or piece level. The DSNU512I (DATASET LEVEL RECOVERY IS REQUIRED) error message indicates that recovery cannot be done at the space level. Recovery must instead be requested at the partition or piece level. Objects with these errors are not recovered and the RECOVER ends with RC8 indicating errors were encountered.

For more information, see [Recovering a data set or partition \(Db2 Utilities\)](#).

REPAIR utility WRITELOG for decompression dictionaries

Function level 100 introduces the capability to write a decompression dictionary log record up to the maximum log record size supported by Db2. This capability can be used after you run an application or utility that builds a new dictionary without writing the old one to the log. So, it is useful for replication products that would otherwise require a refresh of the replication target. After the decompression dictionary is successfully written, the REPAIR utility issues message DSNU3335I with the location of the log record. Applications can then use this information to insert a SYSIBM.SYSCOPY record.

For more information, see the option descriptions for TYPE 'X'4002 and SUBTYPE 'X'000A' in [Syntax and options of the REPAIR control statement \(Db2 Utilities\)](#), and [DSNU3335I \(Db2 Messages\)](#).

Change REORG INDEX SHRLEVEL REFERENCE or CHANGE so the NOSYSUT1 keyword is the default

Starting at function level 500, the NOSYSUT1 keyword is the default for the REORG INDEX utility when specified with the SHRLEVEL REFERENCE or CHANGE keywords. So, the utility avoids use of a work data set, which improves performance and allows REORG INDEX to use parallel subtasks to unload and to build index keys. The default value of the REORG_INDEX_NOSYSUT1 subsystem parameter is also changed from NO to YES, and YES is now the only option. So, this subsystem parameter no longer influences the behavior of REORG INDEX.

For more information, see [Syntax and options of the REORG INDEX control statement \(Db2 Utilities\)](#) and [REORG TS NOPAD DEFAULT \(REORG_TS_NOPAD_DEFAULT subsystem parameter\) \(Db2 Installation and Migration\)](#).

Improved tape data set estimation for LOAD

APAR PH47800 (December 2022) improves size estimates for the LOAD utility when the input is tape data sets managed by DFSMSrmm. A more accurate row estimate results in better sort processing for LOAD PRESORT and better sorting of keys for rebuilding indexes. If you use tape data sets managed by DFSMSrmm as input to LOAD, you do not need to specify the SPACE parameter for the discard and sort input data sets.

FL 504 Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

Additionally, if you use LOAD PRESORT, you no longer need to specify NUMRECS and SORTKEYS. For more information, see [PH47800](#).

Insight, instrumentation, and serviceability in Db2 13

Improved default statistics collection granularity

Starting in function level 100, that default value of the STATIME_MAIN subsystem parameter is changed to 10 seconds. The STATIME_MAIN subsystem parameter specifies the time interval in seconds, for collection of interval-driven statistics that are not collected at the interval that is specified by the STATIME subsystem parameter. With the default statistics interval set at 60 seconds in earlier releases, Db2 database administrators and system programmers cannot identify true workload peaks by using Db2 statistics for subsystem level performance tuning and planning. Similarly, a slowdown of 5 - 15 seconds is difficult to diagnose with statistics collected at a 60-second interval.

See [MAIN STATS TIME field \(STATIME_MAIN subsystem parameter\)](#) (Db2 Installation and Migration)

Index page split instrumentation enhancements

Function level 100 introduces IFCID 396 to provide detailed information about index splits. When data is inserted into a base table, the corresponding indexes are modified accordingly. As a result, performance of SQL insert operations can be impacted during the *index split* process, where synchronous I/O is required under data sharing and the group buffer pool GBP is dependent on the related index page sets.

The existing IFCID 359 records already contain information index split events. However, the information that is recorded is not detailed enough to identify the cause of performance issues. IFCID 359 is also disabled by default, and it can miss capturing some abnormal index split situations.

396 is always enabled by default under statistics trace class 3 and performance trace class 6. An IFCID 396 record is generated when an index split is considered an abnormal split process, such as when the total elapsed time is greater than 1 second. The generated IFCID 396 record contains information such as the DBID, PSID, member ID, URID, page number, and more. The information is helpful for both customers and IBM Support to identify the root cause of INSERT performance issues.

For more information, see the IFCID 396 descriptions in Chapter 11, “IFCID changes in Db2 13,” on page 111 and in the [Trace field descriptions](#) (Db2 Performance).

Starting after the catalog level V13R1M501 update (which requires function level 500), the following RTS columns in the SYSIBM.SYSINDEXSPACESTATS catalog table are populated. They record and aggregate general index split information since last table reorganization, index rebuild, or load replace:

Column name	Data type	Description
REORGTOTALSPLITS	INTEGER	The number of index splits since last reorganization or rebuild.
REORGSPPLITIME	BIGINT	Aggregated elapsed time for all index splits since last reorganization or rebuild.
REORGEXCSPLITS	INTEGER	The number of abnormal index splits (such as elapsed times greater than 1 second) since last reorganization or rebuild.

Db2 starts populating these RTS columns as soon as the catalog level V13R1M501 update completes, even before function level 501 is activated.

Accounting information on the longest wait times for common suspension types

When Db2 transactions take a long time, it is important to determine:

- Where the transaction time is spent.
- Whether the problem is many short suspensions or a few long suspensions.
- Which resources the suspensions are for.

Before this enhancement, detailed performance traces were required to find this information. This enhancement simplifies the diagnosis process by providing information in Db2 accounting records for

the longest wait time for a number of common suspension types. The following suspension types are included:

- IRLM lock suspensions
- Db2 internal latch suspensions
- Waits for Db2 synchronous or asynchronous I/O
- Waits for Db2 service tasks
- Waits for page latches

For more information, see [Chapter 11, “IFCID changes in Db2 13,” on page 111](#) and the DSNWMSGs file.

Long names support for timeout and deadlock messages in IRLM

Starting at function level 500, Db2 13 introduces IRLM support for long names for client information such as workstation ID, user ID, and transaction ID, in deadlock and timeout messages DSNT175I and DSNT376I.

In Db2 12 and earlier, the long name values are truncated in the message output.

With this change, Db2 13 also starts populating existing long name fields in IFCID 172 and IFCID 196 records. These fields remain unpopulated in Db2 12 and earlier.

Partition range support in IFCID 306 for users of replication applications

Function level 100 introduces the capability for applications that collect IFCID 306 trace records for the log read process to request filtering on a range of partitions. For more information, see the new WQLSFLG flag and WQLSDBPP mapping in [Qualification fields for READS requests \(Db2 Performance\)](#).

EDITPROC support in IFCID 306 for users of nonproxy mode replication applications

Function level 100 introduces the support for any user of IFCID 306 in non-proxy mode to use EDITPROC support as an on-request function. For more information, see the new values X'04', X'05', X'06', and X'07' for the 1-byte WQALLOPT field in [Qualification fields for READS requests \(Db2 Performance\)](#).

Improved DBAT status information in DISPLAY THREAD output

APAR PH45504 (July 2022) improves Db2 reports the status of database access threads (DBATs), so that you can distinguish between DBATs that are processing SQL requests for active transactions and threads that reached a transaction boundary and are waiting for new transaction requests. Specifically, this APAR introduces the new status values that are listed in the following table in DISPLAY THREAD command output and in IFCID148 data obtained in IFC READS calls.

Before this APAR, Db2 returns the RA status in all such cases, which makes it difficult to accurately determine the current state of DBATs that use KEEP_DYNAMIC(YES) or high-performance DBAT support. That is, you cannot distinguish between threads that are processing SQL requests for active transactions and threads that reached a transaction boundary and those that are waiting for new transaction requests.

ST value	Meaning
RD	The DBAT is processing a request from a remote location and waiting for the next transaction. The DBAT is an active KEEP_DYNAMIC-refreshable thread, and not a candidate for idle thread detection and termination. For more information, see Enabling KEEP_DYNAMIC refresh for DBATs (Db2 Performance) .
RH	The DBAT is processing a request from a remote location and waiting for the next transaction. The DBAT is being kept active due to resources of various types being held past a commit or rollback. A DBAT with this status is a candidate for idle thread detection and termination based on the IDTHTOIN subsystem parameter setting. For more information, see IDLE THREAD TIMEOUT field (IDTHTOIN subsystem parameter) (Db2 Installation and Migration) .

ST value	Meaning
RP	The DBAT is processing a request from a remote location and waiting for the next transaction. The thread is an active high-performance DBAT, and it is not a candidate for idle thread detection and termination. For more information, see Enabling high-performance DBATs (Db2 Performance) .

FL 504 Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [Managing distributed database access threads \(DBATs\) \(Db2 Performance\)](#)
- [DSNV041I](#)
- [PH44504](#)

Improved DBAT status for MONITOR THREADS profiles in DISPLAY THREAD output

FL 500 Starting in Db2 13 at function level 500 or higher, improves the status values in DISPLAY THREAD output for DBATs that are queued because the MAXDBAT subsystem parameter or the except threshold for a MONITOR THREADS profile was reached. This APAR introduces the new status value RS in DSNV402I to indicate a thread that is suspended because a MONITOR THREADS profile was reached. The existing RQ value is also updated to indicate that it only applies that threads that are suspended because the MAXDBAT value was reached.

This APAR also adds a new counter in the output for the DISPLAY DDF command with the DETAIL option. The new PQDBAT counter in message DSNL093I indicates the current number of DBATs queued because a system profile exception threshold was reached.

FL 504 Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [ST in DSNV401I](#)
- [PQDBAT in DSNL080I](#)
- [PH47626](#)

Discovery and granular control for secure TLS connectivity

APAR PH48764 (March 2023) introduces the capability to discover and enforce the use of approved authentication and encryption methods by Db2 clients. It introduces a set of actions that you can specify in the DSN_PROFILE_ATTRIBUTES_TABLE.KEYWORDS column. These actions use the pattern MONITOR *product-type* CONNECTIONS FOR SECURITY, where *product-type* is one of the following:

- REST
- JDBC
- CLI
- DB2CONNECT
- DSN
- * (the asterisk is a wildcard that applies for application requesters that do not match any of the more specific product types above)

The new keyword values can only be specified for profiles using the default location filtering criteria (DSN_PROFILE_TABLE - LOCATION column contains '*', '::0', or '0.0.0.0' representing any IP address). These new keyword values enable the definition of profiles, based on the application requester product type, to discover and enforce the usage of authorization mechanisms and encrypted connections.

FL 504 Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [Discovering and controlling secure connectivity with profile tables \(Managing Security\)](#)
- [DSN_PROFILE_ATTRIBUTES profile table \(Db2 Performance\)](#)
- [DSNT775I \(Db2 Messages\)](#)
- [DSNT776I \(Db2 Messages\)](#)
- [PH48764](#)

Increased capacity for Db2 trace: More OP buffers

Starting in function level 504, the maximum number of OP buffers for trace monitoring with READA calls is increased from 8 to 16. The increase in the number of available OP buffers can have the following effects:

- More monitoring capacity is available when multiple applications start traces with multiple OP destinations.
- Monitoring applications can experience better performance when they send trace data to a larger number of destinations.

For more information, see the following related topics:

- [Requesting data asynchronously from a monitor program \(Db2 Performance\)](#)
- [READA \(Db2 Performance\)](#)
- [-START TRACE \(Db2\) \(Db2 Commands\)](#)

APAR PH55622 delivered the functional code for increasing the capacity for Db2 trace.

Product identifier (PRDID) values for specific DRDA levels

APAR PH48184 (December 2022) enhances product identifier (PRDID) values returned by Db2, so that they accurately reflect the DRDA level of the Db2 for z/OS data server, which corresponds to a specific Db2 13 function level. As before, the PRDID value is an 8-byte character value in *pppvrrm* format, where: *ppp* is a 3-letter product code; *vv* is the version; *rr* is the release; and *m* is the modification level. With this APAR in Db2 13, the modification level (0 - 9 or A - Z) indicates a specific function level. For example:

DSN13012 for V13R1M501.
DSN13011 for V13R1M500.
DSN13010 for V13R1M100.

Before this APAR and in Db2 12, the PRDID values indicate only a range of function levels. For example, the following values are used in Db2 12 uses these values:

DSN12015 for V12R1M500 or higher.
DSN12010 for V12R1M100.

[FL 504](#) Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [Product identifier \(PRDID\) values in Db2 for z/OS \(Db2 Administration Guide\)](#)
- [PH48184](#)

Improved index traverse count information in DISPLAY STATS output

APAR PH51565 (January 2023) improves how Db2 13 reports index traverse count information in the TRAV. COUNT field of message DSNT380I when you issue a DISPLAY STATS command with the INDEXTRAVERSECOUNT keyword. It also adds a new FTB FACTOR field in DSNT380I, which returns a calculated value that Db2 uses to identify indexes that might benefit from the fast index traversal (FTB) capability.

Before this APAR, the TRAV. COUNT field contains the calculated FTB factor value, which can sometimes become negative, such as when index page splits occur or index-lookaside is used.

When the DISPLAY STATS command output in DSNT380I returns information about multiple indexes, they are now returned in descending order starting with the index with the highest FTB factor value.

IFCID 389 is also updated to display the FTB factor values.

FL 504 Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [-DISPLAY STATS \(Db2\) \(Db2 Commands\)](#)
- [DSNT830I \(Db2 Messages\)](#)
- [Fast index traversal \(Db2 Performance\)](#)
- [PH51565](#)

Latch monitoring and serviceability enhancements

Starting in Db2 12 with APAR PH41756 (July 2023), the messages that Db2 issues when it boosts the priority of a long-waiting latch holder are updated to include additional information about the latch that is causing a suspension. Specifically, the DSNV491I message, which is issued with DSNV523I, is updated with field labels for all values that it returns. New fields are also added for token and ACE values, which can help you obtain client information, such as the end user id, transaction name, and workstation name.

The updated messages have the following format:

```
NAME ST A REQ ID AUTHID PLAN ASID TOKEN
-----
text t t num text text text hex num
V491-LATCH=latch-addr LATCH ASID=latch-asid latch-status JOBNAME=hldr-jobname ASID=
hldr-asid CORRID=hldr-corr-id TOKEN=token-number ACE=ace-num
V490-SUSPENDED timestamp smodule soffset smodptf-level
```

Before this APAR, the DSNV491I message already included information such as the latch holder job name, latch ASID, and whether the latch is held or released but blocked by another holder. However, because it was intended for internal use by IBM Support these fields were not labeled in the message output or described in the documentation.

FL 504 Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [DSNV523I \(Db2 Messages\)](#)
- [DSNV491I \(Db2 Messages\)](#)
- [PH41756](#)

More granular accounting trace records for distributed applications

Starting in Db2 12 with APAR PH55241 (October 2023), you can request that Db2 generates more granular accounting trace records for distributed database applications, by issuing a MODIFY DDF command with the ACCTG(ALLCOMMIT) option. This option also requires that the PTF for APAR OA61811 is applied in your z/OS environment. When this option is set, Db2 generates an accounting trace record occurrence after any commit or rollback. However, Db2 still maintains the WLM enclave that was created by Db2, prior to the initial requested transaction which left open resources. Also, Db2 maintains information so that when the DBAT is terminated after completing the work requests from its client connection, it deletes the enclave and informs WLM of the transactions that were completed.

To use the ACCTG(ALLCOMMIT) option, you must also set the CMTSTAT subsystem parameter value to INACTIVE.

When you issue a MODIFY DDF command with the ACCTG(ALLCOMMIT) option, Db2 issues the following message:

```
DSNL302I ACCTG IS SET TO ALLCOMMIT
```

To return to the existing default behavior, you can issue the following MODIFY DDF command, which includes the ACCTG keyword with no option value.

```
-MODIFY DDF ACCTG
```

When you issue this command Db2 issues the following message:

```
DSNL302I ACCTG IS SET TO COMMIT_WITH_NO_OPEN_RESOURCES
```

Tip: If you specify the ACCTG(ALLCOMMIT) option, but the PTF for APAR OA61811 is not applied in z/OS, or if the CMTSTAT subsystem parameter is set to ACTIVE, Db2 also issues the preceding message to indicate that the default behavior is still being used.

For existing distributed threads, the specified ACCTG option takes effect only after the thread generates an accounting trace record at a commit or rollback with no open resources. Until then, the previously saved ACCTG option for the thread continues to control when it generates accounting trace records.

When this option is set back to its default setting, or if the setting was never changed, Db2 generates an accounting trace record when no open resource exists past a commit or rollback. If any of the following resources remain open and pending, Db2 does not generate an accounting trace record:

- Open WITH HOLD cursors
- Declared global temporary tables that are not dropped
- Unreleased LOB locators
- A commit processed by a stored procedure when running with 2-phase commit rules
- Allocated RELEASE DEALLOCATE packages (due to the -MODIFY DDF PKGREL(BNDOPT) option)
- Allocated KEEP DYNAMIC YES packages

However, if only one of the last two preceding resource types, but not both, are open after a commit or rollback, Db2 generates an accounting trace record occurrence after a commit or rollback.

Important: When the ACCTG(ALLCOMMIT) setting is used, the volume of accounting trace record occurrences increases. The size of the increase depends on whether distributed application workloads have a large number of the above open resource types. When no resources are left open after a commit or rollback, -MODIFY DDF ACCTG(ALLCOMMIT) has essentially no effect. You can also use the following subsystem parameters to control the value and size of the accounting trace records:

- When set to a numeric value, the [ACCUMACC](#) value controls the volume of trace record occurrences.
- When set to ON, the [SMFCOMP](#) value controls size of accounting trace records.

The DISPLAY DDF command output displays the ACCTG setting in the DSNL106I message only if the ACCTG(ALLCOMMIT) setting is in effect.

For more information, see the following related topics:

- [-MODIFY DDF \(Db2\) \(Db2 Commands\)](#)
- [PH55241](#)
- [OA61811](#)

Improved processing for active log full message DSNJ110E

Starting in Db2 13 with APAR PH56288 (December 2023, when the last available active log data set is 5% or more full, Db2 keeps the last DSNJ110E message highlighted on the console until the condition is resolved. The goal is to prevent operators from overlooking this message.

For more information, see the following related topics:

- [DSNJ110E \(Db2 Messages\)](#)
- [Recovering from being out of space in active logs \(Db2 Administration Guide\)](#)
- [PH56288](#)

STATIME_DDF support

Starting in Db2 12 with APAR PH56228 (December 2023), the new subsystem parameter STATIME_DDF controls the interval of location statistics trace records, IFCIDs 365, 411, and 412. All three traces now allow a time interval of up to 5 minutes.

For more information, see the following related topics:

- [DDF STATS TIME field \(STATIME_DDF subsystem parameter\) \(Db2 Installation and Migration\)](#)
- [PH56228](#)

More granular filtering for monitoring secure connectivity with profiles

APAR PH57811 (January 2024) introduces a capability to apply security profile rules more precisely. This capability is especially useful for enforcing security for new cloud-based clients or specific portions of the network more strictly. With this APAR applied, you can specify any of the following values in the LOCATION column for profiles that use the MONITOR *product-type* CONNECTIONS FOR SECURITY keyword:

- '*', '::0', or '0.0.0.0' (for all connections).
- A domain name that resolves to an IP address. An example fully qualified domain name is 'stlmvs1.svl.example.com'.
- IPV4 or IPV6 IP address.
- IPV4 or IPV6 subnet address.

Before this APAR, only the following values are supported for such profiles: '*', '::0', or '0.0.0.0'.

For more information, see the following related topics:

- [Discovering and controlling secure connectivity with profile tables \(Managing Security\)](#)
- [DSN_PROFILE_TABLE profile table \(Db2 Performance\)](#)
- [DSN_PROFILE_ATTRIBUTES profile table \(Db2 Performance\)](#)
- [PH57811](#)

Hardware and software requirements for Db2 13

Db2 13 operates with the following 64-bit architecture IBM zSystems hardware and software:

Hardware:	IBM System zEC12 or later
Operating system:	z/OS 2.4 or later
Other software:	IRLM 2.3 running at function level 3.050 or later
	You can use the following command to determine the IRLM function level: <code>MODIFY irlmproc,STATUS,ALLI</code>

For information about system requirements for Db2 13 for z/OS, see the [Program Directories for Db2 13](#).

Related concepts

[Changes to plan for in Db2 13](#)

Use this information when you are planning migration to Db2 13 and for planning to adopt new capabilities that Db2 13 introduces.

Related reference

[Db2 13 function levels](#)

New Db2 capabilities and enhancements are continuously delivered in a single maintenance stream as the code becomes ready. You can activate the new capabilities in a data sharing group or Db2

subsystem after a function level is delivered. A *function level* corresponds to a single PTF that enables the activation of a specific set of enhancements that shipped in previous prerequisite or co-requisite PTFs. The activation of a function level results in the activation of all lower function levels.

New-function APARs for Db2 13

New capabilities are introduced to Db2 13 through continuous delivery of APARs in a single service stream. Many APARs in Db2 13 deliver deactivated functional code to support future function levels. However, some APARs introduce their enhancements in Db2 13 regardless of the activated function level.

Chapter 2. New-function APARs for Db2 13

New capabilities are introduced to Db2 13 through continuous delivery of APARs in a single service stream. Many APARs in Db2 13 deliver deactivated functional code to support future function levels. However, some APARs introduce their enhancements in Db2 13 regardless of the activated function level.

The following table summarizes the new-function APARs for Db2 13 from newest to oldest, including the APARs that enable activation of Db2 13 function levels. APARs with new-function code for future function levels are listed with the APAR that enables activation of each function level. Only new-function APARs that have PTFs for Db2 13 are included. The following table does not include "bug fixes" or other maintenance APARs.

Tip: Depending on when and how you order the Db2 13 product code, you might find that external changes from any of the following APARs are already built-in when you install or migrate to Db2 13. Also, depending on your maintenance strategy, external changes from APARs that you did not apply in Db2 12 are likely to be already built-in when you migrate to Db2 13. See the descriptions of APARs with availability dates earlier than 2022-06 in [New-function APARs for Db2 12](#).

If a *verifying function level* is listed for an APAR, it is a PREREQ for the enabling APAR for the function level. As a result, activation of the indicated function level verifies that the PTF is applied in Db2 13, including all data sharing members. The verifying function level does not control new-function availability for these APARs. The new function can be used in each data sharing member when the PTF is applied in that member.

Description	APAR	Date	Verifying function level "1" on page 36
"More granular filtering for monitoring secure connectivity with profiles" on page 36	PH57811	2024-01	
"Detail information for a page with minimum LRSN that causes a GRECP recovery delay" on page 36	PH54199	2024-01	
"Improved processing for active log full message DSNJ110E" on page 37	PH56288	2023-12	
"STATIME_DDF support" on page 37	PH56228	2023-12	
Function level 504 activation support <ul style="list-style-type: none">• "Minimized impact of invalidated packages with statement-level invalidation" on page 56 (PH55497)• "Increased capacity for Db2 trace: More OP buffers" on page 57 (PH55622)• "Utility object-level history" on page 57 (PH55476, PH55914, PH55915, and PH55916)• "Enhancements for SQL Data Insights" on page 57 (PH55212)• "ORDER BY support for a fullselect that invokes LISTAGG" on page 57 (PH55596)	PH54919	2023-10	
"Fast index traversal (FTB) controlling rows in SYSIBM.SYSINDEXCONTROL are read more frequently by Db2" on page 37	PH57071	2023-10	

Description	APAR	Date	Verifying function level “1” on page 36
“Display privilege authority for DISPLAY PROFILE and DISPLAY DYNQUERYCAPTURE” on page 37	PH56996 PH56997	2023-10	
“BIND QUERY restriction removed for certain SQL PL packages” on page 38	PH54056	2023-10	
“Specify MQ administration IDs in DSNTIJRT/DSNTRIN” on page 38	PH55025	2023-09	
“More granular accounting trace records for distributed applications” on page 39	PH55241	2023-08	
“Dynamic election of SQL Data Insights vector prefetch on tables” on page 40	PH55212	2023-08	FL 504
“View management authorization enhancements” on page 40	PH54863, PH54936	2023-07	FL 504
“Latch monitoring and serviceability enhancements” on page 41	PH41756	2023-07	FL 504
“Freeing inactive packages for native REST services” on page 41	PH54129	2023-06	FL 504
“Migration readiness reported in DISPLAY GROUP command output” on page 42	PH50072	2023-06	FL 504
“Enhanced SQLDA output for LOBs and distinct types in REXX applications” on page 43	PH52852	2023-04	FL 504
“Generating tailored JCL jobs in batch mode for migrating to Db2 13 or activating function levels” on page 43	PH52482	2023-04	FL 504
“Discovery and granular control for secure TLS connectivity” on page 44	PH48764	2023-03	FL 504
“APPLCOMPAT and PLANMGMT support in DSNTIJRT/DSNTRIN for bind or rebind of packages for Db2-supplied routines ” on page 45	PH50704	2023-03	FL 504
“New vector prefetch capability and improved built-in scalar functions of SQL Data Insights” on page 46	PH51892	2023-02	FL 504
Function level 503 activation support <ul style="list-style-type: none"> • “New default values in existing rows for added ROW CHANGE TIMESTAMP columns” on page 60 (PH51185) • “Accelerator-only support for more than 32K elements in an IN list” on page 60 (PH50756) • “SELECT INTO statement support for OPTIMIZE FOR n ROWS” on page 61 (PH50010) 	PH51469	2023-02	
“Improved index traverse count information in DISPLAY STATS output” on page 46	PH51565	2023-01	FL 504
“Installation and migration changes to support new EXPLAIN columns” on page 47	PH48053	2023-01	FL 504

Description	APAR	Date	Verifying function level “1” on page 36
“Improved IFCID 0402 reset behavior” on page 47	PH50729	2023-01	FL 504
“Removed stacking limitations for PBG to PBR conversions” on page 47	PH51359	2022-12	FL 504
“Control the maximum number of concurrent user-defined external scalar functions” on page 48	PH44833	2022-12	FL 504
“Lock avoidance for singleton SELECT with ISOLATION(CS) and CURRENTDATA(YES)” on page 48	PH49335	2022-12	FL 504
“Authorization update for CREATE INDEX on DGTT” on page 49	PH48601	2022-11	FL 504
“Product identifier (PRDID) values for specific DRDA levels ” on page 49	PH48184	2022-11	FL 504
“Accelerator expression-offload support for LISTAGG and RAND ” on page 50	PH48480	2022-11	FL 504
“Improved tape data set estimation for LOAD” on page 49	PH47800	2022-10	FL 504
“Improved PLAN_TABLE.PARENT_PLANNO column data for views and table expressions” on page 50	PH49972	2022-10	FL 504
Function level V13R1M502 activation support <ul style="list-style-type: none"> • “Db2 controlled sysplex workload balancing ” on page 64 (PH48253) • “Record dependencies and validity at the statement level for packages” on page 65 (PH47560) 	PH47891	2022-10	
“List prefetch for MERGE statements” on page 50	PH47581	2022-09	FL 504
“Support for UNI_90 locale for the LOWER, TRANSLATE, and UPPER built-in functions” on page 51	PH47187	2022-09	FL 504
“Reduced LOGREC entries for parallel queries with FETCH FIRST n ROWS” on page 51	PH48183	2022-09	FL 504
“Improved DBAT status for MONITOR THREADS profiles in DISPLAY THREAD output” on page 51	PH47626	2022-08	FL 504
“DECFLOAT datatype support for PL/I and C with DCLGEN” on page 52	PH47453	2022-08	FL 504
“Improved DBAT status information in DISPLAY THREAD output” on page 52	PH45504	2022-07	FL 504
“New serviceability message for LOAD FORCE and REORG FORCE” on page 53	PH44941	2022-06	FL 504
“New LOAD option to reclaim NPI space” on page 53	PH46346	2022-05	FL 504
“Improvements to the -START ML and -STOP ML commands ” on page 53	PH46031	2022-05	FL 504
New statistics trace classes for monitoring distributed user statistics	PH44815	2022-04	FL 504

Notes

1. Activating the *verifying function level* verifies that the PTF is applied on every Db2 13 data sharing member. However, new function in the APAR can be used in each member when the PTF is applied on that member.

New-function APARs for Db2 13 in 2024

More granular filtering for monitoring secure connectivity with profiles

APAR PH57811 (January 2024) introduces a capability to apply security profile rules more precisely. This capability is especially useful for enforcing security for new cloud-based clients or specific portions of the network more strictly. With this APAR applied, you can specify any of the following values in the LOCATION column for profiles that use the MONITOR *product-type* CONNECTIONS FOR SECURITY keyword:

- '*', '::0', or '0.0.0.0' (for all connections).
- A domain name that resolves to an IP address. An example fully qualified domain name is 'stlmvs1.svl.example.com'.
- IPV4 or IPV6 IP address.
- IPV4 or IPV6 subnet address.

Before this APAR, only the following values are supported for such profiles: '*', '::0', or '0.0.0.0'.

For more information, see the following related topics:

- [Discovering and controlling secure connectivity with profile tables \(Managing Security\)](#)
- [DSN_PROFILE_TABLE](#) profile table (Db2 Performance)
- [DSN_PROFILE_ATTRIBUTES](#) profile table (Db2 Performance)
- [PH57811](#)

Detail information for a page with minimum LRSN that causes a GRECP recovery delay

Starting in Db2 12 with APAR PH54199 (January 2024), adds a new DSNB360I message for detailed information for a page that is identified with minimum page LRSN in a DSNB355I message, to help you locate the page and the object it belongs to promptly in the case the GBP recovery delay is caused by the lagging minimum page LRSN.

Recovery from the group buffer pool recovery pending (GRECP) state might be delayed when the GBP-recovery LRSN is not progressing. The GBP-recovery LRSN is the minimum of the minimum page LRSN and the minimum member LRSN. The minimum page LRSN is the oldest changed page clean-to-dirty LRSN that was recorded at the time of the last group buffer pool checkpoint. The minimum member LRSN is the minimum member-level, write-pending LRSN that was recorded at the last group buffer pool checkpoint.

If the minimum page LRSN is older than the time when the third-to-last group buffer pool checkpoint was taken or the minimum member LRSN is invalid, Db2 issues a DSNB355I message at a group buffer pool checkpoint. The DSNB355I message includes the minimum page LRSN and the local timestamp of the minimum page-level LRSN, but it does not include the page details.

For more information, see the following related topics:

- [DSNB360I \(Db2 Messages\)](#)
- [DSNB355I \(Db2 Messages\)](#)
- [PH54199](#)

New-function APARs for Db2 13 in 2023

Improved processing for active log full message DSNJ110E

Starting in Db2 13 with APAR PH56288 (December 2023, when the last available active log data set is 5% or more full, Db2 keeps the last DSNJ110E message highlighted on the console until the condition is resolved. The goal is to prevent operators from overlooking this message.

For more information, see the following related topics:

- [DSNJ110E \(Db2 Messages\)](#)
- [Recovering from being out of space in active logs \(Db2 Administration Guide\)](#)
- [PH56288](#)

STATIME_DDF support

Starting in Db2 12 with APAR PH56228 (December 2023), the new subsystem parameter STATIME_DDF controls the interval of location statistics trace records, IFCIDs 365, 411, and 412. All three traces now allow a time interval of up to 5 minutes.

For more information, see the following related topics:

- [DDF STATS TIME field \(STATIME_DDF subsystem parameter\) \(Db2 Installation and Migration\)](#)
- [PH56228](#)

Function level V13R1M504 activation support

PH54919 (October 2023) introduces support for activating function level 504 in Db2 13. For more information, see the following related topics:

- [“Function level 504 \(APAR PH54919 - October 2023\)” on page 55](#)
- [PH54919](#)

Fast index traversal (FTB) controlling rows in SYSIBM.SYSINDEXCONTROL are read more frequently by Db2

PH57071 (October 2023) increases how frequently Db2 reads newly inserted rows for control of fast index traversal (FTB) processing in the SYSIBM.SYSINDEXCONTROL table. With this APAR, Db2 applies the newly inserted rows every 2 minutes, instead of every 10 minutes.

For more information, see the following related topics:

- [Enabling or disabling fast index traversal at the index level \(Db2 Performance\)](#)
- [SYSINDEXCONTROL catalog table \(Db2 SQL\)](#)
- [PH57071](#)

Display privilege authority for DISPLAY PROFILE and DISPLAY DYNQUERYCAPTURE

APARs PH56996 and PH56997 (October 2023) introduce support for users with the DISPLAY privilege to issue the DISPLAY PROFILE and DISPLAY DYNQUERYCAPTURE commands. This change makes these commands more consistent with the behavior for other DISPLAY commands.

Before these APARs, users issuing these commands needed to have at least one of the following administrative authorities:

- SQLADM authority
- System DBADM authority
- SYSOPR authority
- SYSCTRL authority
- SYSADM authority

For more information, see the following related topics:

- [-DISPLAY DYNQUERYCAPTURE \(Db2\) \(Db2 Commands\)](#)

- [-DISPLAY PROFILE \(Db2\) \(Db2 Commands\)](#)
- [System privileges \(RACF Access Control Module Guide\)](#)
- [PH56996](#)
- [PH56997](#)

BIND QUERY restriction removed for certain SQL PL packages

APAR PH54056 (October 2023) relaxes a restriction against a BIND QUERY command that is issued to enforce statement-level plan hints for certain SQL PL packages.

Before this APAR, Db2 issues message DSNT281I with reason code 5 when processing a statement that has a DSN_USER_QUERY_TABLE row with a TYPE='N' column value. This restriction prevents the enforcement of statement-level access path hints for statements in SQL PL packages, which have the TYPE='N' value in the SYSPACKAGE catalog table.

For more information, see the following related topics:

- [Specifying access paths at the statement level \(Db2 Performance\)](#)
- [BIND QUERY \(DSN\) \(Db2 Commands\)](#)
- [PH54056](#)

Specify MQ administration IDs in DSNTIJRT/DSNTRIN

Starting in Db2 13 function level V13R1M100 or higher with APAR PH55025 (September 2023), you can specify a list of authorization IDs to be granted ALL privileges on the Db2 MQ tables SYSIBM.MQPOLICY_TABLE and SYSIBM.MQSERVICE_TABLE when you install or migrate Db2. A new optional configuration (DB2OPT) keyword called MQ_ADMIN_ID is added to installation job DSNTIJRT for this purpose. A new MQ ADMIN ID(s) field is also added on panel DNSTIPG1 for setting this parameter. This change helps to support appropriate separation of duties between users of MQ functions and administrators of MQ policies and services.

Before this APAR, when DSNTIJRT/DSNTRIN generates the access list for the two MQ tables based on all the authorization IDs specified in the GRANTTO parameter of each Db2 MQ function, and the result is that any user with EXECUTE privilege on any Db2 MQ function also has ALL privileges on the two Db2 MQ tables. Thus, any user that is allowed to read from or write messages into MQ queues through Db2 MQ functions is also able to delete or change entries in SYSIBM.MQPOLICY_TABLE and SYSIBM.MQSERVICE_TABLE.

This new DSNTIJRT/DSNTRIN optional configuration DB2OPT parameter serves two purposes when specified:

- Specifying a list of authorization IDs that will be granted ALL privileges on the Db2 MQ tables.
- Specifying that only the SELECT privilege on the MQ tables will be granted to all users with EXECUTE privilege on any of the Db2 MQ functions.

The following example shows how to specify previous behavior in MQ_ADMIN_ID parameter setting in job DSNTIJRT:

```
//DSNTRIN EXEC PGM=DSNTRIN,COND=(4,LT),
//          PARM=('Db2SSN(!DSN!) MODE(INSTALL)',
//          ' AUTHID(!AUTHID!) SECDEFID(!SECDEFID!)',
//          ' DEFPGOWN(!DEFPKOWN!)')
//
//DB2OPT DD *
//STOGROUP(SYSDEFLT)
//INDEXSTOG(SYSDEFLT)
//BP4K(BP0)
//BP8K(BP8K0)
//BP16K(BP16K0)
//BP32K(BP32K)
//LOBBP8K(BP8K0)
//LOBBP16K(BP16K0)
//LOBBP32K(BP32K)
//IMS_SECURITY(Db2)
//MQ_SECURITY(Db2)
//SOAP_SECURITY(Db2)
//RTN_PKG_APPLCOMPAT(DEFAULT)
```

```
RTN_PKG_PLANMGMT(DEFAULT)  
MQ_ADMIN_ID(DEFAULT)
```

The following example shows how to specify a list of authorization IDs in the MQ_ADMIN_ID parameter to get the new behavior:

```
MQ_ADMIN_ID(MQUID1,MQUID2,MQUID3)
```

For more information, see the following related topics:

- [Job DSNTIJRT \(Db2 Installation and Migration\)](#)
- [MQ ADMIN ID\(S\) field \(Db2 Installation and Migration\)](#)
- [DSNTIPG1: Installation preferences panel 2 \(Db2 Installation and Migration\)](#)
- [Db2 MQ tables \(Db2 Application programming and SQL\)](#)
- [PH55025](#)

More granular accounting trace records for distributed applications

Starting in Db2 12 with APAR PH55241 (October 2023), you can request that Db2 generates more granular accounting trace records for distributed database applications, by issuing a MODIFY DDF command with the ACCTG(ALLCOMMIT) option. This option also requires that the PTF for APAR OA61811 is applied in your z/OS environment. When this option is set, Db2 generates an accounting trace record occurrence after any commit or rollback. However, Db2 still maintains the WLM enclave that was created by Db2, prior to the initial requested transaction which left open resources. Also, Db2 maintains information so that when the DBAT is terminated after completing the work requests from its client connection, it deletes the enclave and informs WLM of the transactions that were completed.

To use the ACCTG(ALLCOMMIT) option, you must also set the CMTSTAT subsystem parameter value to INACTIVE.

When you issue a MODIFY DDF command with the ACCTG(ALLCOMMIT) option, Db2 issues the following message:

```
DSNL302I ACCTG IS SET TO ALLCOMMIT
```

To return to the existing default behavior, you can issue the following MODIFY DDF command, which includes the ACCTG keyword with no option value.

```
-MODIFY DDF ACCTG
```

When you issue this command Db2 issues the following message:

```
DSNL302I ACCTG IS SET TO COMMIT_WITH_NO_OPEN_RESOURCES
```

Tip: If you specify the ACCTG(ALLCOMMIT) option, but the PTF for APAR OA61811 is not applied in z/OS, or if the CMTSTAT subsystem parameter is set to ACTIVE, Db2 also issues the preceding message to indicate that the default behavior is still being used.

For existing distributed threads, the specified ACCTG option takes effect only after the thread generates an accounting trace record at a commit or rollback with no open resources. Until then, the previously saved ACCTG option for the thread continues to control when it generates accounting trace records.

When this option is set back to its default setting, or if the setting was never changed, Db2 generates an accounting trace record when no open resource exists past a commit or rollback. If any of the following resources remain open and pending, Db2 does not generate an accounting trace record:

- Open WITH HOLD cursors
- Declared global temporary tables that are not dropped
- Unreleased LOB locators

- A commit processed by a stored procedure when running with 2-phase commit rules
- Allocated RELEASE DEALLOCATE packages (due to the -MODIFY DDF PKGREL(BNDOPT) option)
- Allocated KEEP DYNAMIC YES packages

However, if only one the last two preceding resources types, but not both, are open after a commit or rollback, Db2 generates an accounting trace record occurrence after a commit or rollback.

Important: When the ACCTG(ALLCOMMIT) setting is used, the volume of accounting trace record occurrences increases. The size of the increase depends on whether distributed application workloads have a large number of the above open resource types. When no resources are left open after a commit or rollback, -MODIFY DDF ACCTG(ALLCOMMIT) has essentially no effect. You can also use the following subsystem parameters to control the value and size of the accounting trace records:

- When set to a numeric value, the [ACCUMACC](#) value controls the volume of trace record occurrences.
- When set to ON, the [SMFCOMP](#) value controls size of accounting trace records.

The DISPLAY DDF command output displays the ACCTG setting in the DSNL106I message only if the ACCTG(ALLCOMMIT) setting is in effect.

For more information, see the following related topics:

- [-MODIFY DDF \(Db2\) \(Db2 Commands\)](#)
- [PH55241](#)
- [OA61811](#)

Dynamic election of SQL Data Insights vector prefetch on tables

APAR PH55212 (August 2023) enhances the SQL Data Insights vector prefetch capability by making the enablement decision dynamic based on data and AI cache availability. When the MXAIDTCACH parameter is set to a value between 1 and 512 and a query invokes a SQL DI function on a table, Db2 dynamically elects vector prefetching or row-by-row processing to optimize CPU usage for the query. If needed, Db2 automatically disables vector prefetching for the SQL DI function in the query. This update also delivers Db2 support of numeric data types for the AI_ANALOGY function.

[FL 504](#) Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [Upgrading SQL DI](#)
- [MAX AI DATA CACHING field \(MXAIDTCACH subsystem parameter\) \(Db2 Installation and Migration\)](#)
- [AI_ANALOGY \(Db2 SQL\)](#)

View management authorization enhancements

APARs PH54863 and PH54936 (July 2023) enhance Db2 13 at function level 100 or higher to allow database administrators (DBAs) with DBADM authority on databases to drop a view created for another user and select from a view without always requiring system level authority. APAR PH54936 supports the view management authorization enhancements when using RACF access control authorization exit for access control. With these APARs, DROP statements are allowed for a view if the privilege set includes DBADM authority on the database that contains one of the base tables and the subsystem parameter DBACRVW is set to YES. SELECT statements for a view are allowed if the privilege set includes the required privileges such as SELECT or EXECUTE on all the base objects.

[FL 504](#) Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [DROP \(Db2 SQL\)](#)
- [Authorization for queries \(Db2 SQL\)](#)
- [View privileges \(RACF Access Control Module Guide\)](#)

- [DBADM \(Managing Security\)](#)
- [PH54863](#)
- [PH54936](#)

Latch monitoring and serviceability enhancements

Starting in Db2 12 with APAR PH41756 (July 2023), the messages that Db2 issues when it boosts the priority of a long-waiting latch holder are updated to include additional information about the latch that is causing a suspension. Specifically, the DSNV491I message, which is issued with DSNV523I, is updated with field labels for all values that it returns. New fields are also added for token and ACE values, which can help you obtain client information, such as the end user id, transaction name, and workstation name.

The updated messages have the following format:

```
NAME ST A REQ ID AUTHID PLAN ASID TOKEN
-----
text t t num text text text hex num
V491-LATCH=latch-addr LATCH ASID=latch-asid latch-status JOBNAME=hldr-jobname ASID=
hldr-asid CORRID=hldr-corr-id TOKEN=token-number ACE=ace-num
V490-SUSPENDED timestamp smodule soffset smodptf-level
```

Before this APAR, the DSNV491I message already included information such as the latch holder job name, latch ASID, and whether the latch is held or released but blocked by another holder. However, because it was intended for internal use by IBM Support these fields were not labeled in the message output or described in the documentation.

FL 504 Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [DSNV523I \(Db2 Messages\)](#)
- [DSNV491I \(Db2 Messages\)](#)
- [PH41756](#)

Freeing inactive packages for native REST services

Starting in Db2 13 with APAR PH54129 (June 2023), you can issue the DSN FREE PACKAGE command to free inactive packages for native REST services.

Db2 native REST services are created by invoking the REST service manager API with a `createService` request, or by using the DSN BIND SERVICE command. This inserts a row into the REST services table defining the service and creates an application package. You can then rebind the application package for a REST service with various options by using the DSN REBIND PACKAGE command. During rebind, inactive package copies, such as the original and previous copies, can be generated based on the PLANMGMT option. Phased-out copies of packages can be generated if already running threads have the package allocated when the rebind occurs. Invalid package copies can occur when a dependent object is changed or dropped. These types of inactive packages can build up over time when many rebinds are done.

Before this APAR, the only method of freeing inactive REST service packages is by invoking the REST service manager API with a `dropService` request, or by use of the FREE SERVICE command. The shortcoming to doing this is that the REST service itself is freed and all packages, both active and inactive, are freed. Also the FREE PACKAGE command allows the freeing of inactive packages only if they are not REST services, and is blocked for REST services with error message DSNT246I.

Use of the FREE PACKAGE command to free an active REST service package continues to be blocked with the DSNT264I message.

FL 504 Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [Freeing inactive packages for Db2 REST service \(Db2 REST services\)](#)
- [FREE PACKAGE \(DSN\) \(Db2 Commands\)](#)
- [FREE SERVICE \(DSN\) \(Db2 Commands\)](#)
- [DSNT264I \(Db2 Messages\)](#)
- [PH54129](#)

Migration readiness reported in DISPLAY GROUP command output

Starting in Db2 12 with APAR PH50072 (June 2023), you can determine whether your Db2 12 data sharing group or standalone Db2 subsystem is ready for migration to Db2 13 by issuing a DISPLAY GROUP command with the DETAIL keyword.

Before this APAR, it can be difficult to determine whether all Db2 data sharing members are at sufficient code level and function level, and have the fallback SPE APAR applied.

The new migration readiness report in the DISPLAY GROUP output now indicates whether your environment is ready for migration to the next Db2 release. If Db2 is not ready for migration, the report also includes reason information.

For example, the following migration readiness report indicates that a Db2 12 subsystem has a sufficient code level and the fallback SPE is applied, but it is not ready to migrate to Db2 13 because the highest activated function level is not the last Db2 12 function level:

```

DSN7100I  -DB2A DSN7GCM
*** BEGIN DISPLAY  OF GROUP(.....) CATALOG LEVEL(V12R1M500)
                                CURRENT FUNCTION LEVEL(V12R1M100)
                                HIGHEST ACTIVATED FUNCTION LEVEL(V12R1M100)
                                HIGHEST POSSIBLE FUNCTION LEVEL(V12R1M501)
                                PROTOCOL LEVEL(2)
                                GROUP ATTACH NAME(....)

-----
DB2      SUB      DB2      SYSTEM      IRLM
MEMBER   ID  SYS  CMDPREF  STATUS   LVL      NAME      SUBSYS  PRLMPR21
-----
.....   0 DB2A  -DB2A    ACTIVE   121510  UTEC6     PR21     PRLMPR21
-----
MIGRATION READINESS REPORT
-----
DB2      CODE      SPE      MIGRATION
MEMBER   LEVEL     APAR    ELIGIBLE
-----
.....   V12R1M510  PH37108  YES
-----
MIGRATION READINESS STATUS: SUBSYSTEM IS NOT READY FOR DB2 13
REASON: HIGHEST ACTIVATED FUNCTION LEVEL NOT V12R1M510
-----

```

The GET_CONFIG stored procedure is also updated to display the migration readiness information.

FL 504 Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

Tip: A follow-on APAR, PH58761 improves how the migration readiness status is displayed for Db2 13 members in coexistence situations. For more information, see [PH58761](#).

For more information, see the following related topics:

- [Check readiness for migration to Db2 13 \(Db2 Installation and Migration\)](#)
- [-DISPLAY GROUP \(Db2\) \(Db2 Commands\)](#)
- [DSN7100I \(Db2 Messages\)](#)
- [GET_CONFIG stored procedure \(Db2 SQL\)](#)
- [PH50072](#)

Enhanced SQLDA output for LOBs and distinct types in REXX applications

Starting in Db2 12 with APAR PH52852 (April 2023), if a REXX application issues a PREPARE into statement that includes a LOB column or distinct type, USING BOTH is specified, and the column is defined with labels, Db2 now correctly doubles the SQLD value in the SQLDA, and returns previously missing information, including the CCSID of a LOB column and the name for a distinct type.

Before this APAR, if a REXX application that issues a PREPARE INTO which includes a LOB column, the following errors occur in the SQLDA for REXX applications:

- The CCSID of the LOB is not stored in the SQLCCSID.
- If USING BOTH and the columns are defined with LABELS, the SQLD is not doubled where the labels are not returned in the second set after the columns.
- If a distinct type exists, the name is not returned in SQLTNAME.

FL 504 Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [The REXX SQLDA \(Db2 SQL\)](#)
- [DESCRIBE OUTPUT \(Db2 SQL\)](#)
- [PH52852](#)

Generating tailored JCL jobs in batch mode for migrating to Db2 13 or activating function levels

Starting in Db2 13 with APAR PH52482 (April 2023), you can run a batch process to avoid working the Db2 installation CLIST panels in interactive mode when you need to generate tailored JCL jobs and input files for the following activities:

- [Migrating to Db2 13.](#)
- [Activating Db2 13 function levels.](#)

This APAR introduces a new JCL job named DSNTIJBC, which you can run with current and valid input files to generate the required JCL jobs and input files. To use this batch job, you must customize a supplied parameter override file and provide the following input files:

- A current and valid DSNTIDXx defaults input member for your Db2 subsystem.
- The DSNTIDXA member for target Db2 release.

The DSNTIJBC batch job included two steps:

- Executes the DSNTIFMT program, which formats member DSNTINST, which is the Db2 installation CLIST, into a new member named DSNTINSB that can be run in batch mode.
- Runs the ISPSTART command to run DSNTINSB in batch mode.

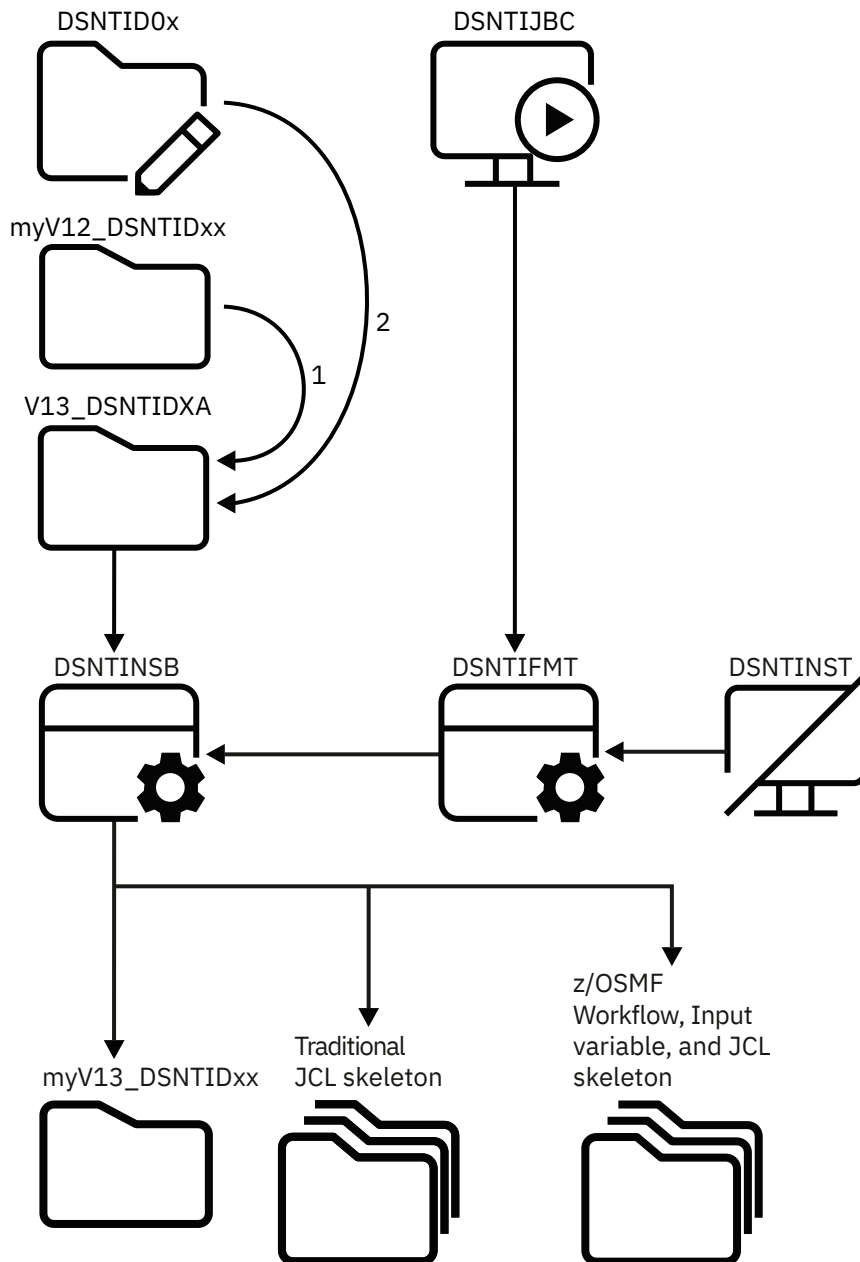


Figure 1. A diagram that illustrates how the DSNTIJBC job uses valid input files to run the Db2 installation CLIST as a background process

FL 504 Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [Generating tailored Db2 migration or function level activation jobs in the background \(Db2 Installation and Migration\)](#)
- [PH52482](#)

Discovery and granular control for secure TLS connectivity

APAR PH48764 (March 2023) introduces the capability to discover and enforce the use of approved authentication and encryption methods by Db2 clients. It introduces a set of actions that you can specify in the DSN_PROFILE_ATTRIBUTES_TABLE.KEYWORDS column. These actions use the pattern *MONITOR product-type CONNECTIONS FOR SECURITY*, where *product-type* is one of the following:

- REST

- JDBC
- CLI
- DB2CONNECT
- DSN
- * (the asterisk is a wildcard that applies for application requesters that do not match any of the more specific product types above)

The new keyword values can only be specified for profiles using the default location filtering criteria (DSN_PROFILE_TABLE - LOCATION column contains '*', '::0', or '0.0.0.0' representing any IP address). These new keyword values enable the definition of profiles, based on the application requester product type, to discover and enforce the usage of authorization mechanisms and encrypted connections.

FL 504 Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [Discovering and controlling secure connectivity with profile tables \(Managing Security\)](#)
- [DSN_PROFILE_ATTRIBUTES profile table \(Db2 Performance\)](#)
- [DSNT775I \(Db2 Messages\)](#)
- [DSNT776I \(Db2 Messages\)](#)
- [PH48764](#)

APPLCOMPAT and PLANMGMT support in DSNTIJRT/DSNTRIN for bind or rebind of packages for Db2-supplied routines

Starting in Db2 12 with APAR PH50704 (March 2023), you can specify the APPLCOMPAT and PLANMGMT bind options when you bind or rebind packages for Db2-supplied routines by running job DSNTIJRT, which executes the DSNTRIN program.

To ensure that all packages for Db2-supplied routines are bound with the same user-specified APPLCOMPAT level, if a valid package whose CONTOKEN matches that of the current DBRM is found, instead of bypassing the package as it currently does, DSNTIJRT/DSNTRIN is updated to perform an additional check to ensure that the APPLCOMPAT level of the package matches that of the user-specified APPLCOMPAT and PLANMGMT bind option. If the APPLCOMPAT levels do not match, DSNTIJRT/DSNTRIN rebinds the packages by using the user-specified APPLCOMPAT and PLANMGMT bind options.

Before this APAR, DSNTRIN only binds the package for a routine when any of the following conditions is true:

- Package does not exist for current DBRM
- Package exists for current DBRM but has been invalidated. Also, the BIND statements generated and executed by DSNTRIN do not specify an APPLCOMPAT parameter, so the APPLCOMPAT bind option defaults to the APPLCOMPAT subsystem parameter setting.

You can get the same behavior as before APAR PH50704 by specifying the DEFAULT options or omitting the new options from DSNTIJRT/DSNTRIN.

Important: If RTN_PKG_APPLCOMPAT(DEFAULT) is specified or applied by default, the APPLCOMPAT subsystem parameter is used to bind the packages. If the APPLCOMPAT subsystem parameter is V10R1 and DSNTIJRT/DSNTRIN has to bind a down-level DSNADMEM package, the bind fails.

FL 504 Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [PKG APPLCOMPAT field \(Db2 Installation and Migration\)](#)
- [PKG PLANMGMT field \(Db2 Installation and Migration\)](#)

- [PH50704](#)

New vector prefetch capability and improved built-in scalar functions of SQL Data Insights

APAR PH51892 (February 2023) introduces a new vector prefetch capability and improves the existing built-in scalar `AI_SEMANTIC_CLUSTER` function, which together with other optimizations, delivers significant functional and performance improvements for SQL Data Insights (SQL DI).

Vector prefetch is the advanced process that SQL DI uses to upload numeric vectors to the IBM Z® AI Optimization Library (zAIO) for calculating similarity scores. The existing query processing architecture forces Db2 to submit one record at a time to z/OS for processing. Prefetching enables Db2 AI functions to submit multiple vectors in a batch at a time, which greatly accelerates SQL DI query processing.

Db2 stores training vectors in a normalized form for all scoring requests. However, the `AI_SEMANTIC_CLUSTER` function requires non-normalized vectors when it computes the average of the input vectors. To meet this requirement, Db2 adds a new `NORMALIZE_ABSOLUTE_VALUE` column to the vector table and populates it with an absolute value during model training. The `AI_SEMANTIC_CLUSTER` function uses the value in the new column to convert the normalized vectors to non-normalized, calculate the input average, and convert the non-normalized vectors back to normalized. The addition of the new column helps improve not only the scoring accuracy of the `AI_SEMANTIC_CLUSTER` function but also the overall scoring performance of the SQL DI feature.

FL 504 Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [AI_SEMANTIC_CLUSTER \(Db2 SQL\)](#)
- [PH51469](#)

Function level V13R1M503 activation support

APAR PH51469 (February 2023) introduces support for activating function level 503 in Db2 13. For more information, see the following related topics:

- [“Function level 503 \(APAR PH51469 - February 2023\)” on page 60](#)
- [PH51469](#)

Improved index traverse count information in DISPLAY STATS output

APAR PH51565 (January 2023) improves how Db2 13 reports index traverse count information in the `TRAV. COUNT` field of message `DSNT380I` when you issue a `DISPLAY STATS` command with the `INDEXTRAVERSECOUNT` keyword. It also adds a new `FTB FACTOR` field in `DSNT380I`, which returns a calculated value that Db2 uses to identify indexes that might benefit from the fast index traversal (FTB) capability.

Before this APAR, the `TRAV. COUNT` field contains the calculated FTB factor value, which can sometimes become negative, such as when index page splits occur or index-lookaside is used.

When the `DISPLAY STATS` command output in `DSNT380I` returns information about multiple indexes, they are now returned in descending order starting with the index with the highest FTB factor value.

IFCID 389 is also updated to display the FTB factor values.

FL 504 Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [-DISPLAY STATS \(Db2\) \(Db2 Commands\)](#)
- [DSNT830I \(Db2 Messages\)](#)
- [Fast index traversal \(Db2 Performance\)](#)
- [PH51565](#)

Installation and migration changes to support new EXPLAIN columns

Starting in Db2 12, APAR PH48053 (January 2023) updates several installation and migration jobs to support converting EXPLAIN tables to a new format that is supported in Db2 12 at function level 501 or higher. The new EXPLAIN table columns (APAR PH44421) support integration of EXPLAIN table data with data used by IBM Db2 AI for z/OS (Db2ZAI) to support the SQL optimization capabilities of Db2ZAI.

FL 504 Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [Chapter 10, “EXPLAIN table changes,” on page 109](#)
- [Migration step 24: Convert EXPLAIN tables to the current format \(Db2 Installation and Migration\)](#)
- [Activating Db2 13 function level 500 or higher \(Db2 Installation and Migration\)](#)
- [PH48053](#)
- [PH44421](#)

Improved IFCID 0402 reset behavior

Starting in Db2 12 or later with APAR PH50729 (March 2023), Db2 resets IFCID 402 statistics counters immediately if the warning or exception threshold for the associated profiles is changed when you issue a START PROFILE command.

In Db2 12 the new behavior is used only if the RESET_IFCID402 option is specified for the DDF_COMPATIBILITY subsystem parameter.

If the new behavior was used in Db2 12, or Db2 13 is installed with the PTF for APAR PH50729 applied, Db2 13 always uses the new behavior.

However, in the following situations, the new behavior takes effect in Db2 13 only after you resubmit your customized copy of DSNTIJUZ and issue a -SET SYSPARM command or stop and restart Db2:

- The PTF for APAR PH50729 was applied in Db2 12 but the RESET_IFCID402 option was not specified for DDF_COMPATIBILITY subsystem parameter.
- Installation or migration to Db2 13 was completed before the PTF for APAR PH50729 is applied.

FL 504 Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [Chapter 4, “Incompatible changes in Db2 13,” on page 89](#)
- [PH50729](#)

New-function APARs for Db2 13 in 2022

Removed stacking limitations for PBG to PBR conversions

Starting in function level 500 or higher with APAR PH51359 (December 2022), Db2 13 supports *stacking* of certain pending data definition changes when a table in a partition-by-growth (PBG) table space is converted to partition-by-range (PBR). That is, the pending definition changes in following table can now be issued together and materialized by a single execution of the REORG utility. This capability is especially useful if you need to enlarge the partition data set sizes to accommodate the distribution of data into the partitions, alter the columns to be used as partitioning keys, or alter other table space or index attributes.

Object level	Supported stacked pending definition changes for PBG to PBR conversion
Table space	<ul style="list-style-type: none"> • BUFFERPOOL • DSSIZE • SEGSIZE (excluding conversion to UTS) • MEMBER CLUSTER
Table	<ul style="list-style-type: none"> • ALTER COLUMN • DROP COLUMN
Index	<ul style="list-style-type: none"> • BUFFERPOOL • COMPRESS

Before this change, Db2 issues SQLCODE -20385 if you try to issue any of these alterations when a PBG to PBR conversion is pending, or in the opposite situation, so at least two executions of the REORG utility are required to complete any of these changes if they are needed for the conversion.

[FL 504](#) Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [Converting tables from growth-based to range-based partitions \(Db2 Administration Guide\)](#)
- [Restrictions for pending data definition changes \(Db2 Administration Guide\)](#)
- [-20385 \(Db2 Codes\)](#)
- [PH51359](#)

Control the maximum number of concurrent user-defined external scalar functions

APAR PH44833 (December 2022) introduces a new subsystem parameter to control the number of user-defined external scalar functions that can run concurrently in a Db2 thread. With this APAR in Db2 12 at function level 100 or higher, the MAX_UDF subsystem parameter now specifies the maximum number of user-defined external scalar functions that can run concurrently in a Db2 thread. The default value is 2000.

[FL 504](#) Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [MAX UDFS field \(MAX_UDF subsystem parameter\) \(Db2 Installation and Migration\)](#)
- [PH44833](#)

Lock avoidance for singleton SELECT with ISOLATION(CS) and CURRENTDATA(YES)

Starting in Db2 12 at function level 100 or higher, APAR PH49335 (December 2022) introduces the capability to use lock avoidance for singleton SELECT statements that run with the ISOLATION(CS) and CURRENTDATA(YES) options. A *singleton SELECT* is a cursor SELECT that returns at most one row.

Db2 can use lock avoidance for such SINGLETON SELECT statements if you specify YES for the LA_SINGLESEL_ISOCS_CDY subsystem parameter.

Allowing lock avoidance can significantly lessen locking activity for such singleton SELECT statements. However, occasional false warning or error conditions are also possible during certain small timing windows, such as SQLCODE -811 (more than one row returned) or SQLCODE +100 (no row found). The YES setting for LA_SINGLESEL_ISOCS_CDY should only be used if your environment can tolerate these situations.

For more information, see the following related topics:

- [LA_SINGLESEL_ISOCS_CDY subsystem parameter \(Db2 Installation and Migration\)](#)
- [Lock avoidance \(Db2 Performance\)](#)
- [PH49335](#)

Improved tape data set estimation for LOAD

APAR PH47800 (December 2022) improves size estimates for the LOAD utility when the input is tape data sets managed by DFSMSrmm. A more accurate row estimate results in better sort processing for LOAD PRESORT and better sorting of keys for rebuilding indexes. If you use tape data sets managed by DFSMSrmm as input to LOAD, you do not need to specify the SPACE parameter for the discard and sort input data sets.

FL 504 Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

Additionally, if you use LOAD PRESORT, you no longer need to specify NUMRECS and SORTKEYS. For more information, see [PH47800](#).

Authorization update for CREATE INDEX on DGT

APAR PH48601 (December 2022) removes certain authorization checks for CREATE INDEX statements for creating indexes on declared global temporary tables (DGTTs). Specifically, this enhancement removes the USE check for the buffer pool and storage group when you creating an index on a DGTT, if the specified buffer pool and storage group are the same as the default values associated with the work file database. If the specified values differ from the defaults,

Before this APAR, the USE authorization checks could cause such CREATE INDEX statements to fail. However, this only occurred if the defaults were changed, or use of the default buffer pool and storage group was revoked from public.

FL 504 Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [CREATE INDEX \(Db2 SQL\)](#)
- [Creating declared temporary tables \(Db2 Application programming and SQL\)](#)
- [PH48601](#)

Product identifier (PRDID) values for specific DRDA levels

APAR PH48184 (December 2022) enhances product identifier (PRDID) values returned by Db2, so that they accurately reflect the DRDA level of the Db2 for z/OS data server, which corresponds to a specific Db2 13 function level. As before, the PRDID value is an 8-byte character value in *pppvrrm* format, where: *ppp* is a 3-letter product code; *vv* is the version; *rr* is the release; and *m* is the modification level. With this APAR in Db2 13, the modification level (0 - 9 or A - Z) indicates a specific function level. For example:

DSN13012 for V13R1M501.

DSN13011 for V13R1M500.

DSN13010 for V13R1M100.

Before this APAR and in Db2 12, the PRDID values indicate only a range of function levels. For example, the following values are used in Db2 12 uses these values:

DSN12015 for V12R1M500 or higher.

DSN12010 for V12R1M100.

FL 504 Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [Product identifier \(PRDID\) values in Db2 for z/OS \(Db2 Administration Guide\)](#)
- [PH48184](#)

Accelerator expression-offload support for LISTAGG and RAND

APAR PH48480 (November 2022) introduces offload support from Db2 12 or later to IBM Db2 Analytics Accelerator offload support for the following built-in functions:

- [LISTAGG \(Db2 SQL\)](#)
- [RANDOM or RAND \(Db2 SQL\)](#) with the optional *numeric-expression* seed value

Offload of these functions is supported by IBM Db2 Analytics Accelerator 7.1.9 or later.

To enable offload of these functions, you specify YES in the ENABLE ACCELERATOR SPECIFIC RESULTS field on panel DSNTIP8A. For more information, see [ACCELERATION OPTIONS field \(QUERY_ACCEL_OPTIONS subsystem parameter\)](#) (Db2 Installation and Migration).

[FL 504](#) Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [PH48480](#)

Improved PLAN_TABLE.PARENT_PLANNO column data for views and table expressions

APAR PH49972 (October 2022) enhances how Db2 populates the PARENT_PLANNO column of the PLAN_TABLE for queries that reference views or table expressions.

After a query that contains a view or table expression is explained, the PLAN_TABLE already contains a plan for accessing the view or table expression, and it also contains a child query block that produces the result rows for the view or table expression. However, in some scenarios before this APAR, no linkage exists in the PLAN_TABLE between rows for the plan and rows for child query block. With this APAR applied, Db2 now writes the PLAN_TABLE.PARENT_PLANNO column value for the plan into the same column for rows for a child query block that returns result rows for a view or table expression.

[FL 504](#) Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [PLAN_TABLE \(Db2 Performance\)](#)
- [PH49972](#)

Function level V13R1M502 activation support

APAR PH47891 (October 2022) introduces support for activating function level 502 in Db2 13. For more information, see the following related topics:

- [“Function level 502 \(APAR PH47891 - October 2022\)” on page 63](#)
- [PH47891](#)

List prefetch for MERGE statements

APAR PH47581 (September 2022) introduces list prefetch as a possible access path for the MERGE statement. Index access with list prefetch can sometimes be used when an index column is being updated by the MERGE statement. Prior to this APAR, such MERGE statements would likely resort to table space scans. For more information, see the following related topics:

[FL 504](#) Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

- [Index access for MERGE \(Db2 Performance\)](#)
- [MERGE \(Db2 SQL\)](#)
- [List prefetch \(PREFETCH='L' or 'U'\) \(Db2 Performance\)](#)
- [PH47581](#)

Support for UNI_90 locale for the LOWER, TRANSLATE, and UPPER built-in functions

APAR PH47187 (September 2022) adds support to specify that the LOWER, TRANSLATE, and UPPER built-in functions use z/OS Unicode Conversion Services only with "normal" casing capabilities and use Unicode Standard 9.0.0. Db2 now supports a new locale, UNI_90, which specifies that z/OS Unicode Conversion Services is used with only "normal" casing capabilities and use Unicode Standard 9.0.0.

Before to this APAR, Db2 supports the following locales to indicate that Db2 should use z/OS Unicode Conversion Services to perform the upper or lower conversion with "normal" or "normal" and "special" casing capabilities.

- 'UNI' uses Unicode version 3.0.0 (which is the default for z/OS Unicode Services).
- 'UNI_SIMPLE' uses the latest version of the Unicode Standard that z/OS Unicode Conversion Services supports.
- 'UNI_60' specifies that z/OS Unicode Conversion Services is used with only "normal" casing capabilities and use Unicode Standard 6.0.0.

FL 504 Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [LOWER \(Db2 SQL\)](#)
- [UPPER \(Db2 SQL\)](#)
- [TRANSLATE \(Db2 SQL\)](#)
- [PH47187](#)

Reduced LOGREC entries for parallel queries with FETCH FIRST *n* ROWS

Starting in Db2 13 function level 500 or higher, APAR PH48183 (September 2022) reduces the number of EREP LOGREC entries with reason code 00E50013 for parallel queries that specify FETCH FIRST *n* ROWS.

If a query that specifies FETCH FIRST *n* ROWS runs in parallel, the parallel child tasks continue fetching more records as the parent retrieves and returns them to the application. As soon as *n* rows are returned, the parent sends a "stop" message to the child tasks. When each child receives the stop message, it stops processing as if canceled. Before this APAR, each child task also writes out an EREP LOGREC entry with reason code 00E50013 in this situation.

With APAR PH48183 applied, the parallel child tasks can stop processing sooner and avoid issuing the EREP LOGREC entries with 00E50013.

The new logic applies for applications that run at application compatibility V13R1M500 or higher.

FL 504 Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [00E50013 \(Db2 Codes\)](#)
- [PH48183](#)

Improved DBAT status for MONITOR THREADS profiles in DISPLAY THREAD output

FL 500 Starting in Db2 13 at function level 500 or higher, improves the status values in DISPLAY THREAD output for DBATs that are queued because the MAXDBAT subsystem parameter or the except threshold for a MONITOR THREADS profile was reached. This APAR introduces the new status value RS in DSNV402I to indicate a thread that is suspended because a MONITOR THREADS profile was reached. The existing RQ value is also updated to indicate that it only applies that threads that are suspended because the MAXDBAT value was reached.

This APAR also adds a new counter in the output for the DISPLAY DDF command with the DETAIL option. The new PQDBAT counter in message DSNL093I indicates the current number of DBATs queued because a system profile exception threshold was reached.

FL 504 Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [ST in DSNV401I](#)
- [PQDBAT in DSNL080I](#)
- [PH47626](#)

DECFLOAT datatype support for PL/I and C with DCLGEN

APAR PH47453 (August 2022) enhances DCLGEN to support the DECFLOAT data type for the PL/I and C programming languages.

FL 504 Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [Data types that DCLGEN uses for variable declarations \(Db2 Application programming and SQL\)](#)
- [DCLGEN \(declarations generator\) \(Db2 Application programming and SQL\)](#)
- [PH47453](#)

Improved DBAT status information in DISPLAY THREAD output

APAR PH45504 (July 2022) improves Db2 reports the status of database access threads (DBATs), so that you can distinguish between DBATs that are processing SQL requests for active transactions and threads that reached a transaction boundary and are waiting for new transaction requests. Specifically, this APAR introduces the new status values that are listed in the following table in DISPLAY THREAD command output and in IFCID148 data obtained in IFC READS calls.

Before this APAR, Db2 returns the RA status in all such cases, which makes it difficult to accurately determine the current state of DBATs that use KEEP_DYNAMIC(YES) or high-performance DBAT support. That is, you cannot distinguish between threads that are processing SQL requests for active transactions and threads that reached a transaction boundary and those that are waiting for new transaction requests.

ST value	Meaning
RD	The DBAT is processing a request from a remote location and waiting for the next transaction. The DBAT is an active KEEP_DYNAMIC-refreshable thread, and not a candidate for idle thread detection and termination. For more information, see Enabling KEEP_DYNAMIC refresh for DBATs (Db2 Performance) .
RH	The DBAT is processing a request from a remote location and waiting for the next transaction. The DBAT is being kept active due to resources of various types being held past a commit or rollback. A DBAT with this status is a candidate for idle thread detection and termination based on the IDTHTOIN subsystem parameter setting. For more information, see IDLE THREAD TIMEOUT field (IDTHTOIN subsystem parameter) (Db2 Installation and Migration) .
RP	The DBAT is processing a request from a remote location and waiting for the next transaction. The thread is an active high-performance DBAT, and it is not a candidate for idle thread detection and termination. For more information, see Enabling high-performance DBATs (Db2 Performance) .

FL 504 Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [Managing distributed database access threads \(DBATs\) \(Db2 Performance\)](#)

- [DSNV041I](#)
- [PH44504](#)

New serviceability message for LOAD FORCE and REORG FORCE

Starting in Db2 12 with APAR PH44941 (June 2022), when you specify the FORCE option for the LOAD utility or REORG utility, and the utility forces the cancellation of blocking claimers, the utility job output will now include the new message DSNU2929I. This message reports the class of claimers that are canceled and the time of the cancellation. This information aids in serviceability.

[FL 504](#) Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [Syntax and options of the LOAD control statement \(Db2 Utilities\)](#)
- [Syntax and options of the REORG INDEX control statement \(Db2 Utilities\)](#)
- [Syntax and options of the REORG TABLESPACE control statement \(Db2 Utilities\)](#)
- [DSNU2929I \(Db2 Messages\)](#)
- [PH44941](#)

New LOAD option to reclaim NPI space

APAR PH46346 (or PH39194 in Db2 12) (May 2022) introduces the new LOAD utility option KEEP_EMPTY_PAGES allows you to specify whether LOAD deletes empty index leaf pages of nonpartitioned secondary indexes (NPSIs) when keys are deleted from the logical partitions of the NPSI. This option is applicable to only LOAD SHRLEVEL NONE PART REPLACE utility executions.

The main benefit of keeping these empty pages (KEEP_EMPTY_PAGES=YES) is improved utility performance. Additionally, new index keys that are inserted by LOAD or other applications can reuse the empty index leaf pages. The benefit of deleting these empty pages (KEEP_EMPTY_PAGES=NO) is that space can be reclaimed. Prior to this enhancement, these empty leaf pages were always kept. This behavior continues to be the default if KEEP_EMPTY_PAGES NO is not explicitly specified.

[FL 504](#) Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [Syntax and options of the LOAD control statement \(Db2 Utilities\)](#)
- [PH46346](#)

Improvements to the -START ML and -STOP ML commands

With APAR PH46031 (or PH43479 in Db2 12) (May 2022), the commands to start and stop the Db2 functions used by IBM Db2 AI for z/OS® have been improved to allow for the separate activation or deactivation of either the SQL optimization or the system assessment and distributed connection control features of IBM Db2 AI for z/OS.

[FL 504](#) Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [-START ML \(Db2\) \(Db2 Commands\)](#)
- [-STOP ML \(Db2\) \(Db2 Commands\)](#)
- [PH46031](#)

New statistics trace classes for monitoring distributed user statistics

With APAR PH44815 (April 2022), you can use the following new statistics trace classes to monitor statistics for DRDA connections:

- Statistics class 10 records statistics in IFCID 411 for remote applications based on the CURRENT_CLIENT_APPLNAME special register.
- Statistics class 11 records statistics in IFCID 412 for remote users based on the CURRENT_CLIENT_USERID special register.

For more information, see the following related topics:

- [Preparing to monitor distributed connections \(Db2ZAI\)](#)
- [PH44815](#)

Chapter 3. Db2 13 function levels

New Db2 capabilities and enhancements are continuously delivered in a single maintenance stream as the code becomes ready. You can activate the new capabilities in a data sharing group or Db2 subsystem after a function level is delivered. A *function level* corresponds to a single PTF that enables the activation of a specific set of enhancements that shipped in previous prerequisite or co-requisite PTFs. The activation of a function level results in the activation of all lower function levels.

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!](#)

About function levels in Db2 13

A *function level* enables a particular set of new Db2 capabilities and enhancements that were previously delivered in the single continuous stream of Db2 code. It includes code that supports new capabilities, defect fixes, and preventive service items. Before you can use the new capabilities of a function level, you must activate the function level, or a higher function level. Activation of a function level implies activation of the capabilities that are introduced by all lower function levels.

For more information about function levels, and how to activate them in Db2 13, see [Part 3, “Adopting new capabilities in Db2 13 continuous delivery,” on page 133.](#)

Available function levels

The following function levels are available in Db2 13. They are listed in release order, beginning with the highest available function level.

Function level 504 (APAR PH54919 - October 2023)

Function level 504 introduces minimized impact from invalidated packages, increased Db2 trace capacity with more OP buffers, utility object-level history, enhancements for SQL Data Insights, and ORDER BY support for a *fullselect* that calls the LISTAGG built-in function.

Contents

[“Minimized impact of invalidated packages with statement-level invalidation” on page 56](#)

[“Increased capacity for Db2 trace: More OP buffers” on page 57](#)

[“Utility object-level history” on page 57](#)

[“Enhancements for SQL Data Insights” on page 57](#)

[“ORDER BY support for a fullselect that invokes LISTAGG” on page 57](#)

[“Verification of enhancements to Db2 13 in new-function APARs” on page 58](#)

[“How to activate function level 504” on page 58](#)

Enabling APAR:	PH54919
Full identifier:	V13R1M504
Catalog level required:	V13R1M504
Product identifier (PRDID):	DSN13015

New capabilities in function level 504

Function level 504 activates the following new capabilities in Db2 13.

Minimized impact of invalidated packages with statement-level invalidation

Starting in function level 504, you can enable packages for statement-level invalidation. This new capability uses the statement-level dependency infrastructure, including the new `DEPLEVEL(STATEMENT)` bind option, that was introduced in function level 502. See [“Record dependencies and validity at the statement level for packages”](#) on page 65.

With statement-level dependencies and statement-level invalidation, applications can execute a package that is invalidated at a statement level without waiting for the completion of the automatic rebind (autobind). The application thread can execute the valid statements immediately. Any invalid statements go through incremental bind in the application thread before execution. Concurrently, Db2 initiates a special autobind, which is called *autobind phase-in*. When the autobind phase-in finishes successfully, the subsequent executing threads can use the new valid copy of the package, and they no longer use incremental bind for any statements.

The autobind phase-in process runs concurrently with executing application threads, and the old invalid copy becomes a phased-out copy, which is stored in the `SYSIBM.SYSPACKCOPY` table. The new copy of the package becomes the current copy. Subsequent executions of the package use the new valid copy. Threads that existed prior to autobind phase-in completing can also use the new current copy when they release the phased-out copy (based on the `RELEASE(COMMIT)` or `RELEASE(DEALLOCATE)` bind options). In this respect, autobind phase-in is the same as rebind phase-in. However, unlike other phase-in rebinds, autobind phase-in does not require the `PLANMGMT` subsystem parameter to be set to `EXTENDED`. For more about rebind-phase in, see [Phase-in of package rebinds \(Db2 Application programming and SQL\)](#).

Db2 uses automatic binds only when the `ABIND` subsystem parameter is set to `YES` or `COEXIST`. If `ABIND` is set to `NO` when an invalid package runs, Db2 returns an error. For details, see [AUTO BIND field \(ABIND subsystem parameter\) \(Db2 Installation and Migration\)](#).

Incompatible change: If incremental binds fail due to prior invalidating changes to dependent objects, applications that previously failed with SQL code -904 might start receiving different SQL codes. For example, an incremental bind for a statement that depended on a dropped table might return -204 to the application. For more information, see [Chapter 4, “Incompatible changes in Db2 13,”](#) on page 89.

Note: At function level 502 or higher, a package that is bound with `DEPLEVEL(STATEMENT)` option can be marked with `VALID='S'` if it is invalidated by a `DROP FUNCTION` statement. However, after the PTF for APAR PH55497 is applied, Db2 always marks a package that is invalidated by a `DROP FUNCTION` statement with `VALID='N'`.

For more information, see the following related topics:

- [Autobind phase-in for packages with statement-level invalidation \(Db2 Application programming and SQL\)](#)
- [Enabling autobind phase-in for packages invalidated at the statement level \(Db2 Application programming and SQL\)](#)
- [MAX_CONCURRENT_PKG_OPS in macro DSN6SPRM \(Db2 Installation and Migration\)](#)
- [SYSPACKAGE catalog table \(Db2 SQL\)](#)
- [00E30305 \(Db2 Codes\)](#)

APAR PH55497 delivered the functional code for minimizing the impact of invalidated packages.

Increased capacity for Db2 trace: More OP buffers

Starting in function level 504, the maximum number of OP buffers for trace monitoring with READA calls is increased from 8 to 16. The increase in the number of available OP buffers can have the following effects:

- More monitoring capacity is available when multiple applications start traces with multiple OP destinations.
- Monitoring applications can experience better performance when they send trace data to a larger number of destinations.

For more information, see the following related topics:

- [Requesting data asynchronously from a monitor program \(Db2 Performance\)](#)
- [READA \(Db2 Performance\)](#)
- [-START TRACE \(Db2\) \(Db2 Commands\)](#)

APAR PH55622 delivered the functional code for increasing the capacity for Db2 trace.

Utility object-level history

Starting in function level 504, you can collect object-level information along with utility execution information for IBM Db2 Utilities. Real time and historical information about the table spaces and index spaces for IBM Db2 Utilities executions are inserted to the SYSIBM.SYSOBJEVENTS catalog table.

System administrators and DBAs can use queries to generate, filter, order, and join the information in the SYSIBM.SYSOBJEVENTS and SYSIBM.SYSUTILITIES catalog tables to check, analyze, and compare utility executions.

For more information, see the following related topics:

- [Monitoring utility history \(Db2 Utilities\)](#)
- [Utility object-level history information in SYSOBJEVENTS \(Db2 Utilities\)](#)
- [UTILITY HISTORY \(UTILITY_HISTORY subsystem parameter\) \(Db2 Installation and Migration\)](#)
- [SYSUTILITIES catalog table \(Db2 SQL\)](#)
- [SYSOBJEVENTS catalog table \(Db2 SQL\)](#)

The following APARs delivered the functional code for utility object-level history: PH55476, PH55914, PH55915, and PH55916.

Enhancements for SQL Data Insights

Function level 504 introduces the following enhancements for the SQL DI capability in Db2 13:

- New Db2 built-in AI_COMMONALITY function that computes a similarity score by using the value of the *expression* argument and the centroid value of the model column. You can use the function in your queries to detect the common patterns and the outliers in your data.
- Enhanced AI_ANALOGY function with the numeric data type support. You can specify numeric data types in the arguments for the function.
- Automatic disablement of SQL DI vector prefetching based on the AI object type and the AI cache size. When the MXAIDTCACH parameter is set to a value greater than 0 and a query invokes a SQL DI function on a table, Db2 dynamically chooses between vector prefetching and row-by-row processing to optimize the CPU usage of the function.

APAR PH55212 delivered the functional code for these SQL DI enhancements.

ORDER BY support for a *fullselect* that invokes LISTAGG

Function level 504 removes a restriction against specifying an ORDER BY clause in a fullselect that contains an invocation of the LISTAGG built-in function for applications that run at application compatibility level V13R1M504 or higher. For more information, see [LISTAGG \(Db2 SQL\)](#).

APAR PH55596 delivered the functional code for ORDER BY support for a *fullselect* that invokes LISTAGG.

Verification of enhancements to Db2 13 in new-function APARs

Many APARs introduce enhancements to Db2 13 that are available at any function level. That is, their changes take immediate effect as you apply the PTF to each Db2 data sharing member.

However, starting with function level 504, activating a Db2 13 function level also verifies that the PTFs for a specific set of these previously available APARs are consistently applied in the Db2 13 environment. For more information, see [Chapter 2, “New-function APARs for Db2 13,” on page 33](#).

V13R1M504 application compatibility

Most new SQL capabilities become available only to applications that use the equivalent application compatibility (APPLCOMPAT) level or higher. For a list, see [Chapter 7, “SQL changes in Db2 13,” on page 99](#).

For more information about application compatibility levels, see [Chapter 23, “Controlling the Db2 application compatibility level,” on page 155](#).

How to activate function level 504

The following steps summarize the process for activating this function level. To learn more about how to activate and control the adoption of new capabilities available for use in your Db2 13 environment and continuous delivery in general, see [Part 3, “Adopting new capabilities in Db2 13 continuous delivery,” on page 133](#).

Procedure

To activate function level 504, complete the following steps:

1. If Db2 13 is still at function level 100, activate function level 500 first. For more information, see [Activating Db2 13 function level 500 or higher \(Db2 Installation and Migration\)](#).
2. Generate tailored JCL jobs for the CATMAINT and function level activation steps. You can use the DSNTIJBC batch job or the Db2 installation CLIST.

Tip: You can avoid working through the Db2 installation CLIST panels in interactive mode by running a batch job with valid input files to generate the required JCL jobs and input files with a background process. See [Generating tailored Db2 migration or function level activation jobs in the background \(Db2 Installation and Migration\)](#).

To generate the required JCL jobs and input files with a background process, complete the following steps:

- a. Customize the DSNTIDOA parameter override file by following that instructions in the file.
- b. Customize the DSNTIJBC job. For example, if *prefix*.SDSNSAMP(DSNTIDOA) is the customized parameter override file, you can specify the following values in the IPSTART command in DSNTIJBC.

```
ISPSTART CMD(%DSNTINSB +  
  OVERPARM(prefix.SDSNSAMP(DSNTIDOA)) +  
  ) BREDIMAX(1)
```

- c. If you use Db2 Value Unit Edition, you must also provide the data set name of the DSNTIDVU parameter override file in the IPSTART command in the DSNTIJBC job, as shown in the following example, where *prefix*.SDSNSAMP(DSNTIDVU) is the customized OTC LICENSE file.

```
ISPSTART CMD(%DSNTINSB +  
  OVERPARM(<prefix>.SDSNSAMP(DSNTIDOA)) +  
  OTCLPARM(<prefix>.SDSNSAMP(DSNTIDVU)) +  
  ) BREDIMAX(1)
```

- d. Submit the customized DSNTIJBC job.

To generate the required JCL jobs and input files with the Db2 installation CLIST in interactive mode, complete the following steps:

- a. In panel DSNTIPA1, specify `INSTALL TYPE ===> ACTIVATE`. Then, specify the name of the output member from the previous function level activation (or migration) in the `INPUT MEMBER` field, and specify a new member name in the `OUTPUT MEMBER` field.
 - b. In panel DSNTIP00, specify the current function level and `TARGET FUNCTION LEVEL ===> V13R1M504`. The Db2 installation CLIST uses this value when it tailors the `ACTIVATE` command in the DSNTIJAF job and the `CATMAINT` utility control statement in the DSNTIJTC job.
 - c. Proceed through the remaining Db2 installation CLIST panels, and wait for the Db2 installation CLIST to tailor the jobs for the activation process. The output data set contains the tailored jobs for the activation process. For more information, see [The Db2 installation CLIST panel session \(Db2 Installation and Migration\)](#).
3. Ensure that no incompatible applications will interfere with the catalog update. For details, see [Chapter 19, “Identifying applications that are incompatible with catalog updates,”](#) on page 145.
 4. Update the catalog and verify the changes for function level 504 by completing the following steps:
 - a. Run the tailored DSNTIJTC job, or run the `CATMAINT` utility with `LEVEL V13R1M504`, to update the catalog to the appropriate catalog level. If multiple catalog updates are required, the `CATMAINT` job processes each update in sequential order. If a later update in the sequence fails, the previous successful updates do not roll back, and the catalog level remains at the highest level reached. If that occurs, you can correct the reason for the failure and resubmit the same `CATMAINT` job.
 - b. Run the generated DSNTIJX2 job to run the `CHECK INDEX` utility for Db2 catalog and directory indexes for new objects created in Db2 13.
 5. Check that Db2 is ready for function level activation by issuing the following `ACTIVATE` command with the `TEST` option:

```
-ACTIVATE FUNCTION LEVEL (V13R1M504) TEST
```

Db2 issues message DSNU757I to indicate the results. For more information, see [Chapter 18, “Testing Db2 function level activation,”](#) on page 143.

6. Run the tailored DSNTIJAF job, or issue the following `ACTIVATE` command:

```
-ACTIVATE FUNCTION LEVEL (V13R1M504)
```

7. If you are ready for applications to use new capabilities in this function level, rebind them at the corresponding application compatibility level. For more information, see [Chapter 23, “Controlling the Db2 application compatibility level,”](#) on page 155.

Optionally, when you are ready for all applications to use the new capabilities of the target function level, you can run the following jobs:

- a. Run DSNTIJUZ to modify the subsystem parameter module with the `APPLCOMPAT` value that was specified on panel DSNTIP00.
- b. Run DSNTIJOZ job to issue `SET SYSPARM` command to bring the `APPLCOMPAT` subsystem parameter changes online.
- c. Run DSNTIJUA job to modify the Db2 data-only application defaults module with the `SQLLEVEL` value that was specified on panel DSNTIP00.

Related tasks

[Activating Db2 13 function level 500 or higher \(Db2 Installation and Migration\)](#)

[Activating Db2 13 function levels](#)

You control the activation and adoption of new features in Db2 13 by activating function levels and specifying the application compatibility level. You can also continue to apply corrective and preventative service without adopting new feature function.

Related reference

[-ACTIVATE \(Db2\) \(Db2 Commands\)](#)

Function level 503 (APAR PH51469 - February 2023)

Function level 503 introduces improved default behavior for added ROW CHANGE TIMESTAMP columns in existing rows, accelerator-only support for queries with IN list predicates with more than 32K elements with IBM Db2 Analytics Accelerator V7 for z/OS, and *optimize-clause* support for SELECT INTO statements.

Contents

[“New default values in existing rows for added ROW CHANGE TIMESTAMP columns” on page 60](#)

[“Accelerator-only support for more than 32K elements in an IN list” on page 60](#)

[“SELECT INTO statement support for OPTIMIZE FOR n ROWS” on page 61](#)

[“How to activate function level 503” on page 62](#)

Enabling APAR:	PH51469
Full identifier:	V13R1M503
Catalog level required:	V13R1M501
Product identifier (PRDID):	DSN13014
Incompatible changes:	None

New capabilities in function level 503

Function level 503 activates the following new capabilities in Db2 13.

New default values in existing rows for added ROW CHANGE TIMESTAMP columns

Starting at application compatibility level V13R1M503 or higher, Db2 uses a constant default value when a new ROW CHANGE TIMESTAMP column is added. When processing ALTER TABLE statements that specify ADD COLUMN for ROW CHANGE TIMESTAMP columns, Db2 now sets the corresponding value in the DEFAULTVALUE column value in the SYSIBM.SYSCOLUMNS catalog table to the timestamp of the ALTER TABLE statement. (Note that CREATE TABLE processing to define a ROW CHANGE TIMESTAMP column does not set the DEFAULTVALUE column value.)

Before this change, Db2 derives the default values for existing rows from the page header from the row, which is the RBA for standalone Db2 subsystems. In data sharing, the default is derived is based on an internal mapping table between the LRSN and a timestamp. As a result, inserts, deletes, or updates to any rows in a page can change the derived default row change timestamp column values for unchanged rows, leading to unpredictable results.

For more information, see the following related topics:

- [ROW CHANGE expression \(Db2 SQL\)](#)
- [SYSCOLUMNS catalog table \(Db2 SQL\)](#)
- [ALTER TABLE \(Db2 SQL\)](#)

Starting at function level 503 or higher, redirected recovery processing is also updated to allow different ROW CHANGE TIMESTAMP column default values between source and target tables. For more information, see [Running a redirected recovery \(Db2 Utilities\)](#).

APAR PH51185 delivered the functional code for improve default behavior for added ROW CHANGE TIMESTAMP columns.

Accelerator-only support for more than 32K elements in an IN list

Starting in function level 503, a query that has an IN list of more than 32,767 (32K) elements in an IN predicate can run as an accelerator-only query on IBM Db2 Analytics Accelerator V7 for z/OS, if all of the following conditions are met:

- Query acceleration is enabled and requested for the query. For more information, see [Enabling acceleration of SQL queries \(Db2 Performance\)](#).
- The application runs at Db2 application compatibility level V13R1M503 or higher.
- Option 12 is specified in the list of values for the [QUERY_ACCEL_OPTIONS subsystem parameter](#).
- The target accelerator is IBM Db2 Analytics Accelerator V7 for z/OS.
- The IN list specifies only SQL constants.
- The query can be functionally supported by the target accelerator and IBM Db2 Warehouse.

When such queries run on a V7 accelerator, the enforcement of any limits on the number of elements in the IN list predicate is handled by IBM Db2 Warehouse, instead of Db2 for z/OS. IBM Db2 Warehouse has no documented limit for elements in an IN list. However, practical limitations such as the memory and processing resources available for the query in IBM Db2 Warehouse are still likely to impose some limit on the number of elements that can be specified in the IN list predicate.

A view can also be created in Db2 for z/OS for a query with an IN list that contains more than 32K elements, but only under the same conditions listed previously for queries that contain such IN predicates. Any query that uses such a view is also evaluated and limited to the same conditions and restrictions described above. Such views also have the value 'R' in IBMREQD column in the SYSIBM.SYSVIEWS catalog table, to indicate the Db2 13 release dependency.

For more information, see "Accelerator-only support for more than 32K elements in an IN list" in [IN predicate \(Db2 SQL\)](#).

APAR PH50756 delivered the functional code for the accelerator support for large IN list predicates.

SELECT INTO statement support for OPTIMIZE FOR *n* ROWS

Starting at application compatibility level V13R1M503 or higher, you can specify an *optimize-clause* in SELECT INTO statements to enable Db2 to consider access paths that use a sort. SELECT INTO statements always return a single row. However, you can use OPTIMIZE FOR 2 ROWS can be specified to influence the Db2 optimizer.

When FETCH FIRST 1 ROW ONLY is specified, Db2 also applies a sort avoidance preference that is associated with OPTIMIZE FOR 1 ROWS during access path selection. However, sometimes the avoiding of any sort can result in a more expensive (in total cost) access path being used. If this situation occurs, OPTIMIZE FOR 2 ROWS can be specified in the SELECT INTO statement to enable Db2 to consider use of a more efficient access path that uses a sort. Other integer values can be specified for *n* in OPTIMIZE FOR *n* ROWS, but OPTIMIZE FOR 2 ROWS is recommended for this scenario.

For more information, see the following related topics:

- [SELECT INTO \(Db2 SQL\)](#)
- [optimize-clause \(Db2 SQL\)](#)
- [Interaction between FETCH and OPTIMIZE FOR clauses \(Db2 Performance\)](#)

APAR PH50010 delivered the functional code for *optimize-clause* support for SELECT INTO statements.

V13R1M503 application compatibility

Most new SQL capabilities become available only to applications that use the equivalent application compatibility (APPLCOMPAT) level or higher. For a list, see [Chapter 7, "SQL changes in Db2 13," on page 99](#).

For more information about application compatibility levels, see [Chapter 23, "Controlling the Db2 application compatibility level," on page 155](#).

How to activate function level 503

The following steps summarize the process for activating this function level. To learn more about how to activate and control the adoption of new capabilities available for use in your Db2 13 environment and continuous delivery in general, see [Part 3, “Adopting new capabilities in Db2 13 continuous delivery,” on page 133](#).

Procedure

To activate function level 503, complete the following steps:

1. If Db2 13 is still at function level 100, activate function level 500 first. For more information, see [Activating Db2 13 function level 500 or higher \(Db2 Installation and Migration\)](#).
2. Generate tailored JCL jobs for the CATMAINT and function level activation steps. You can use the DSNTIJBC batch job or the Db2 installation CLIST.

Tip: You can avoid working through the Db2 installation CLIST panels in interactive mode by running a batch job with valid input files to generate the required JCL jobs and input files with a background process. See [Generating tailored Db2 migration or function level activation jobs in the background \(Db2 Installation and Migration\)](#).

To generate the required JCL jobs and input files with a background process, complete the following steps:

- a. Customize the DSNTIDOA parameter override file by following that instructions in the file.
- b. Customize the DSNTIJBC job. For example, if *prefix*.SDSNSAMP(DSNTIDOA) is the customized parameter override file, you can specify the following values in the IPSTART command in DSNTIJBC.

```
ISPSTART CMD(%DSNTINSB +  
  OVERPARM(prefix.SDSNSAMP(DSNTIDOA)) +  
  ) BREDIMAX(1)
```

- c. If you use Db2 Value Unit Edition, you must also provide the data set name of the DSNTIDVU parameter override file in the IPSTART command in the DSNTIJBC job, as shown in the following example, where *prefix*.SDSNSAMP(DSNTIDVU) is the customized OTC LICENSE file.

```
ISPSTART CMD(%DSNTINSB +  
  OVERPARM(<prefix>.SDSNSAMP(DSNTIDOA)) +  
  OTCLPARM(<prefix>.SDSNSAMP(DSNTIDVU)) +  
  ) BREDIMAX(1)
```

- d. Submit the customized DSNTIJBC job.

To generate the required JCL jobs and input files with the Db2 installation CLIST in interactive mode, complete the following steps:

- a. In panel DSNTIPA1, specify `INSTALL TYPE ===> ACTIVATE`. Then, specify the name of the output member from the previous function level activation (or migration) in the INPUT MEMBER field, and specify a new member name in the OUTPUT MEMBER field.
 - b. In panel DSNTIP00, specify the current function level and `TARGET FUNCTION LEVEL ===> V13R1M503`. The Db2 installation CLIST uses this value when it tailors the ACTIVATE command in the DSNTIJAF job and the CATMAINT utility control statement in the DSNTIJTC job.
 - c. Proceed through the remaining Db2 installation CLIST panels, and wait for the Db2 installation CLIST to tailor the jobs for the activation process. The output data set contains the tailored jobs for the activation process. For more information, see [The Db2 installation CLIST panel session \(Db2 Installation and Migration\)](#).
3. Ensure that no incompatible applications will interfere with the catalog update. For details, see [Chapter 19, “Identifying applications that are incompatible with catalog updates,” on page 145](#).
 4. Update the catalog and verify the changes for function level 503 by completing the following steps:
 - a. Run the tailored DSNTIJTC job, or run the CATMAINT utility with `LEVEL V13R1M503`, to update the catalog to the appropriate catalog level. If multiple catalog updates are required, the CATMAINT

job processes each update in sequential order. If a later update in the sequence fails, the previous successful updates do not roll back, and the catalog level remains at the highest level reached. If that occurs, you can correct the reason for the failure and resubmit the same CATMAINT job.

- b. Run the generated DSNTIJX2 job to run the CHECK INDEX utility for Db2 catalog and directory indexes for new objects created in Db2 13.
5. Check that Db2 is ready for function level activation by issuing the following ACTIVATE command with the TEST option:

```
-ACTIVATE FUNCTION LEVEL (V13R1M503) TEST
```

Db2 issues message DSNU757I to indicate the results. For more information, see [Chapter 18, “Testing Db2 function level activation,”](#) on page 143.

6. Run the tailored DSNTIJAF job, or issue the following ACTIVATE command:

```
-ACTIVATE FUNCTION LEVEL (V13R1M503)
```

7. If you are ready for applications to use new capabilities in this function level, rebind them at the corresponding application compatibility level. For more information, see [Chapter 23, “Controlling the Db2 application compatibility level,”](#) on page 155.

Optionally, when you are ready for all applications to use the new capabilities of the target function level, you can run the following jobs:

- a. Run DSNTIJUZ to modify the subsystem parameter module with the APPLCOMPAT value that was specified on panel DSNTIP00.
- b. Run DSNTIJOZ job to issue SET SYSPARM command to bring the APPLCOMPAT subsystem parameter changes online.
- c. Run DSNTIJUA job to modify the Db2 data-only application defaults module with the SQLLEVEL value that was specified on panel DSNTIP00.

Related tasks

[Activating Db2 13 function level 500 or higher \(Db2 Installation and Migration\)](#)

[Activating Db2 13 function levels](#)

You control the activation and adoption of new features in Db2 13 by activating function levels and specifying the application compatibility level. You can also continue to apply corrective and preventative service without adopting new feature function.

Related reference

[-ACTIVATE \(Db2\) \(Db2 Commands\)](#)

Function level 502 (APAR PH47891 - October 2022)

Function level 502 introduces Db2 controlled sysplex workload balancing (WLB) and infrastructure for statement-level invalidation.

Contents

[“Db2 controlled sysplex workload balancing ” on page 64](#)

[“Record dependencies and validity at the statement level for packages” on page 65](#)

[“How to activate function level 502” on page 66](#)

Enabling APAR:	PH47891
Full identifier:	V13R1M502
Catalog level required:	V13R1M501
Product identifier (PRDID):	DSN13013

New capabilities in function level 502

Function level 502 enables the following new capabilities in Db2 13.

Db2 controlled sysplex workload balancing

Starting in function level 502, you can enable sysplex workload balancing (WLB) on a Db2 for z/OS server.

sysplex WLB functionality (also called *transaction-level workload balancing*) on Db2 for z/OS servers provides high availability for client applications that connect directly to a data sharing group by balancing work among members of a data sharing group at the start of a transaction. Db2 for z/OS server returns a list of available members with weights provided by WLM. At the start of a transaction, the client application chooses the member with the best weight from the list. Configuration parameters are available on clients (Java and non-Java) to enable this functionality.

To enable use of this new capability, function level 502 introduces the following new keywords for the `-MODIFY DDF (Db2) (Db2 Commands)` command.

RQSTWLB

Specifies that Db2 returns to the remote client a request to enable sysplex workload balancing. This option can only be specified if the subsystem is part of a data sharing group. This option has group-wide scope. The decision to honor sysplex workload balancing request lies with the client.

Notes:

- The RQSTWLB option applies for Db2 for Linux, UNIX, and Windows 11.5 or later (CLI Db2 code release "SQL11050" and IBM Data Server Driver for JDBC and SQLJ 4.26.14 or later.)
- If RQSTWLB is used, applications might encounter additional communication errors such as SQL codes -30108 and -20542. These errors are removed by default with Db2 11.5.8 or later client drivers, and the client property `enableseamlessfailovererrorcodes` controls whether they are issued.

★

The option is enabled for entire data-sharing group. This is the default option.

location-name

The option is enabled for remote clients that access the data-sharing group using the location-name as the data source.

alias-name

The option is enabled for remote clients that access the data-sharing group using the specified alias-name as data source.

DFTWLB

Db2 honors the sysplex workload balancing option requested by the client. This option can only be specified if the subsystem is part of a data sharing group. This option has group-wide scope. DFTWLB is the default installed option.

★

The option is enabled for entire data-sharing group. This is the default option.

location-name

The option is enabled for remote clients that access the data-sharing group using the location-name as the data source.

alias-name

The option is enabled for remote clients that access the data-sharing group using the specified alias-name as data source.

For more information about enabling sysplex workload balancing with client drivers, see:

- [Java client support for high availability on IBM data servers](#)
- [Non-Java client support for high availability for connections to Db2 for z/OS servers](#)

PH48253 delivered the functional code for Db2 controlled sysplex workload balancing.

Record dependencies and validity at the statement level for packages

Starting in function level 502, when a package is bound or rebound with the new DEPLEVEL(STATEMENT) option, package invalidation is handled at a statement level. If one or more statements in a package are invalidated, a value of 'S' is recorded in the VALID column of the SYSPACKAGE or SYSPACKCOPY catalog table to indicate that the package is partially invalidated. The VALID column of the SYSPACKSTMT or SYSPACKSTMTCOPY catalog tables indicates the specific statement that is invalidated. In function level 503 or lower, a partially invalidated package is handled in the same manner as an invalidated package on REBIND and in an autobind situation.

Tip: [FL 504](#) This enhancement primarily introduces infrastructure to support a new capability that becomes available after you activate function level 504 or higher in Db2 13. For more information see [Enabling autobind phase-in for packages invalidated at the statement level \(Db2 Application programming and SQL\)](#).

The new dependencies recorded are only for statement-level object dependencies. Only changes to an object invalidate a package at the statement level. Any changes in authorization privileges still invalidate a package at the package level. For more information, see [Changes that invalidate packages \(Db2 Application programming and SQL\)](#).

Note: At function level 502 or higher, a package that is bound with DEPLEVEL(STATEMENT) option can be marked with VALID='S' if it is invalidated by a DROP FUNCTION statement. However, after the PTF for APAR PH55497 is applied, Db2 always marks a package that is invalidated by a DROP FUNCTION statement with VALID='N', regardless of the DEPLEVEL bind option for the package.

Package dependencies continue to be stored in the SYSPACKDEP catalog table. If a package is bound with the DEPLEVEL(STATEMENT) option, a new catalog table, SYSPACKSTMTDEP, stores the dependencies between a package statement and the objects referenced by that statement. The SYSPACKSTMTDEP catalog table contains the statement-level dependencies for a current package and all of the package copies.

Incompatible change:

An existing application might be marked with a value of 'S' in the VALID column of the SYSPACKAGE or SYSPACKCOPY table to indicate that the package has individual statements marked as invalid. There are no behavioral differences between how an invalidated package (VALID = 'N') and a partially invalidated package (VALID = 'S') are handled. The VALID value is reset to 'Y' in the same manner. However, if there are jobs or queries that look explicitly for 'N' in this column, these jobs might need to be modified to account for the new value of 'S'.

For more information, see [Chapter 4, “Incompatible changes in Db2 13,” on page 89](#).

For more information, see the following related topics:

- [DEPLEVEL bind option \(Db2 Commands\)](#)
- [PACKAGE DEPENDENCY LEVEL field \(PACKAGE_DEPENDENCY_LEVEL subsystem parameter\) \(Db2 Installation and Migration\)](#)
- [DSNTIP4: Application programming defaults panel 2 \(Db2 Installation and Migration\)](#)
- [SYSPACKSTMT catalog table \(Db2 SQL\)](#)
- [SYSPACKSTMTDEP catalog table \(Db2 SQL\)](#)
- [Changes that invalidate packages \(Db2 Application programming and SQL\)](#)

PH47560 delivered the functional code to record dependencies and validity for a package at the statement level.

V13R1M502 application compatibility

Most new SQL capabilities become available only to applications that use the equivalent application compatibility (APPLCOMPAT) level or higher. For a list, see [Chapter 7, “SQL changes in Db2 13,” on page 99](#).

For more information about application compatibility levels, see [Chapter 23, “Controlling the Db2 application compatibility level,”](#) on page 155.

How to activate function level 502

The following steps summarize the process for activating this function level. To learn more about how to activate and control the adoption of new capabilities available for use in your Db2 13 environment and continuous delivery in general, see [Part 3, “Adopting new capabilities in Db2 13 continuous delivery,”](#) on page 133.

Procedure

To activate function level 502, complete the following steps:

1. If Db2 13 is still at function level 100, activate function level 500 first. For more information, see [Activating Db2 13 function level 500 or higher \(Db2 Installation and Migration\)](#).
2. Generate tailored JCL jobs for the CATMAINT and function level activation steps. You can use the DSNTIJBC batch job or the Db2 installation CLIST.

Tip: You can avoid working through the Db2 installation CLIST panels in interactive mode by running a batch job with valid input files to generate the required JCL jobs and input files with a background process. See [Generating tailored Db2 migration or function level activation jobs in the background \(Db2 Installation and Migration\)](#).

To generate the required JCL jobs and input files with a background process, complete the following steps:

- a. Customize the DSNTIDOA parameter override file by following that instructions in the file.
- b. Customize the DSNTIJBC job. For example, if *prefix*.SDSNSAMP(DSNTIDOA) is the customized parameter override file, you can specify the following values in the IPSTART command in DSNTIJBC.

```
ISPSTART CMD(%DSNTINSB +  
  OVERPARM(prefix.SDSNSAMP(DSNTIDOA)) +  
  ) BREDIMAX(1)
```

- c. If you use Db2 Value Unit Edition, you must also provide the data set name of the DSNTIDVU parameter override file in the IPSTART command in the DSNTIJBC job, as shown in the following example, where *prefix*.SDSNSAMP(DSNTIDVU) is the customized OTC LICENSE file.

```
ISPSTART CMD(%DSNTINSB +  
  OVERPARM(<prefix>.SDSNSAMP(DSNTIDOA)) +  
  OTCLPARM(<prefix>.SDSNSAMP(DSNTIDVU)) +  
  ) BREDIMAX(1)
```

- d. Submit the customized DSNTIJBC job.

To generate the required JCL jobs and input files with the Db2 installation CLIST in interactive mode, complete the following steps:

- a. In panel DSNTIPA1, specify `INSTALL TYPE ===> ACTIVATE`. Then, specify the name of the output member from the previous function level activation (or migration) in the `INPUT MEMBER` field, and specify a new member name in the `OUTPUT MEMBER` field.
 - b. In panel DSNTIP00, specify the current function level and `TARGET FUNCTION LEVEL ===> V13R1M502`. The Db2 installation CLIST uses this value when it tailors the `ACTIVATE` command in the DSNTIJAF job and the CATMAINT utility control statement in the DSNTIJTC job.
 - c. Proceed through the remaining Db2 installation CLIST panels, and wait for the Db2 installation CLIST to tailor the jobs for the activation process. The output data set contains the tailored jobs for the activation process. For more information, see [The Db2 installation CLIST panel session \(Db2 Installation and Migration\)](#).
3. Ensure that no incompatible applications will interfere with the catalog update. For details, see [Chapter 19, “Identifying applications that are incompatible with catalog updates,”](#) on page 145.

4. Update the catalog and verify the changes for function level 502 by completing the following steps:
 - a. Run the tailored DSNTIJTC job, or run the CATMAINT utility with LEVEL V13R1M502, to update the catalog to the appropriate catalog level. If multiple catalog updates are required, the CATMAINT job processes each update in sequential order. If a later update in the sequence fails, the previous successful updates do not roll back, and the catalog level remains at the highest level reached. If that occurs, you can correct the reason for the failure and resubmit the same CATMAINT job.
 - b. Run the generated DSNTIJX2 job to run the CHECK INDEX utility for Db2 catalog and directory indexes for new objects created in Db2 13.

5. Check that Db2 is ready for function level activation by issuing the following ACTIVATE command with the TEST option:

```
-ACTIVATE FUNCTION LEVEL (V13R1M502) TEST
```

Db2 issues message DSNU757I to indicate the results. For more information, see [Chapter 18, “Testing Db2 function level activation,”](#) on page 143.

6. Run the tailored DSNTIJAF job, or issue the following ACTIVATE command:

```
-ACTIVATE FUNCTION LEVEL (V13R1M502)
```

7. If you are ready for applications to use new capabilities in this function level, rebind them at the corresponding application compatibility level. For more information, see [Chapter 23, “Controlling the Db2 application compatibility level,”](#) on page 155.

Optionally, when you are ready for all applications to use the new capabilities of the target function level, you can run the following jobs:

- a. Run DSNTIJUZ to modify the subsystem parameter module with the APPLCOMPAT value that was specified on panel DSNTIP00.
- b. Run DSNTIJOZ job to issue SET SYSPARM command to bring the APPLCOMPAT subsystem parameter changes online.
- c. Run DSNTIJUA job to modify the Db2 data-only application defaults module with the SQLLEVEL value that was specified on panel DSNTIP00.

Related tasks

[Activating Db2 13 function level 500 or higher \(Db2 Installation and Migration\)](#)

[Activating Db2 13 function levels](#)

You control the activation and adoption of new features in Db2 13 by activating function levels and specifying the application compatibility level. You can also continue to apply corrective and preventative service without adopting new feature function.

Related reference

[-ACTIVATE \(Db2\) \(Db2 Commands\)](#)

Function level 501 (Db2 13 installation or migration - May 2022)

Function level 501 (V13R1M501) is the first opportunity after migration to Db2 13 for applications to use new features and capabilities that depend on catalog changes in Db2 13.

Contents

[“New capabilities in function level 501” on page 68](#)

[“How to activate function level 501” on page 69](#)

Enabling APAR:	None
Full identifier:	V13R1M501
Catalog level required:	V13R1M501

Product identifier (PRDID):	DSN13012
Incompatible changes:	None

New capabilities in function level 501

Function level 501 activates the following new capabilities in Db2 13.

Allow applications to specify a deadlock resolution priority

Function level 501 introduces the SYSIBMADM.DEADLOCK_RESOLUTION_PRIORITY built-in global variable to allow an application to specify a priority to use when resolving a deadlock situation with other threads. When an application sets and uses this built-in global variable (by using a *SET assignment-statement* SQL statement), the Db2 subsystem uses that value as a relative weighting factor to resolve deadlock situations with other threads.

For more information, see [DEADLOCK_RESOLUTION_PRIORITY \(Db2 SQL\)](#).

You can also use Db2 profile tables to specify values for the new SYSIBMADM.DEADLOCK_RESOLUTION_PRIORITY built-in global variable for both remote and local applications. See [Setting built-in global variables by using profile tables \(Db2 Administration Guide\)](#).

Profile table enhancements for application environment settings

Db2 13 introduces the capability to use system profiles for local applications in certain situations. Previously, the initial values for special registers and system built-in global variables can be specified in the Db2 profile tables, but they are used only for initialization with distributed threads. The new Db2 profile table support for local applications requires Db2 to be started with the DDF subsystem parameter set to AUTO or COMMAND. See [DDF STARTUP OPTION field \(DDF subsystem parameter\) \(Db2 Installation and Migration\)](#).

Starting in function level 501, Db2 profile tables can be used for both local and remote applications in the following situations:

- You can specify values for the new SYSIBMADM.DEADLOCK_RESOLUTION_PRIORITY built-in global variable. See [DEADLOCK_RESOLUTION_PRIORITY \(Db2 SQL\)](#) and [Setting built-in global variables by using profile tables \(Db2 Administration Guide\)](#).

Real-time statistics scalability

As data volumes become larger, the widths of some columns in the real-time statistics tables are not large enough to accommodate larger values. In addition, during high volume processing, lock escalation might occur on the real-time statistics history table spaces. Lock escalation can negatively affect concurrency and performance.

Starting after function level 501 is activated, the following changes to the real-time statistics tables and table spaces are introduced to provide greater capacity and concurrency:

- In real-time statistics tables SYSIBM.SYSTABLESPACESTATS and SYSIBM.SYSINDEXSPACESTATS, and their associated history tables SYSIBM.SYSTABSPACESTATS_H and SYSIBM.SYSIXSPACESTATS_H, some column data types are BIGINT instead of INTEGER, or INTEGER instead of SMALLINT.
- Lock escalation is disabled on the following table spaces: DSNDB06.SYSTSTSS and DSNDB06.SYSTSISS for the RTS tables; and DSNDB06.SYSTSTSH and DSNDB06.SYSTSISH for the RTS history tables

For more information, see [SYSTABLESPACESTATS catalog table \(Db2 SQL\)](#) and [SYSINDEXSPACESTATS catalog table \(Db2 SQL\)](#).

This enhancement optimizes Db2 13 and improves its performance without changing how you configure, monitor, and use Db2.

Collection of real-time and historical information about utility execution

To improve utility management, function level 501 introduces the ability to collect real-time and historical information about utility execution. After you activate utility history collection by setting the UTILITY_HISTORY subsystem parameter to UTILITY, information about utilities is added to the

SYSIBM.SYSUTILITIES catalog table. One row is inserted into the SYSUTILITIES table at the start of each utility execution. Then, information in the row is updated as the utility progresses, and final information is updated in the row when the utility execution finishes.

For more information, see [Monitoring utility history \(Db2 Utilities\)](#), [UTILITY HISTORY \(UTILITY_HISTORY subsystem parameter\) \(Db2 Installation and Migration\)](#), and [SYSUTILITIES catalog table \(Db2 SQL\)](#).

Real-time statistics support for index splits

Catalog level V13R1M501 introduces three new real-time statistics (RTS) table columns in the Db2 catalog to provide detailed information about index splits.

The following RTS table fields are added in the SYSIBM.SYSINDEXSPACESTATS catalog table to record and aggregate general index split information since the last table reorganization, index rebuild or load replace:

Column name	Data type	Description
REORGTOTALSPLITS	INTEGER	The number of index splits since last reorganization or rebuild.
REORGSPPLITIME	BIGINT	Aggregated-elapsed time for all index splits since last reorganization or rebuild.
REORGECSPLITS	INTEGER	The number of abnormal index splits (such as elapsed times greater than 1 second) since last reorganization or rebuild.

Db2 starts populating these RTS columns as soon as the catalog level V13R1M501 update completes, even before function level 501 is activated.

V13R1M501 application compatibility

Most new SQL capabilities become available only to applications that use the equivalent application compatibility (APPLCOMPAT) level or higher. For a list, see [Chapter 7, “SQL changes in Db2 13,” on page 99](#).

For more information about application compatibility levels, see [Chapter 23, “Controlling the Db2 application compatibility level,” on page 155](#).

How to activate function level 501

In new Db2 13 installations, function level 501 is already activated. Use this procedure if you are migrating from Db2 12.

Before you begin

Before you activate function level 500 or higher, complete the following prerequisite tasks:

1. In Db2 12, identify and resolve incompatible changes and activate function level 510 (V12R1M510). You can run the pre-migration job DSNTIJPE in Db2 12 to identify the incompatible changes. For more information, see [Verify Db2 13 premigration activities and activate function level 510 in Db2 12 \(Db2 Installation and Migration\)](#).

2. Verify that every member was restarted with the fallback SPE applied in Db2 12.

Important: Inactive members that never started in Db2 12 with the fallback SPE ([APAR PH37108](#)) applied cannot start after the first data sharing member is migrated to Db2 13 at function level 100.

3. Migrate the Db2 subsystem or data sharing group to Db2 13, as described in [Migrating your Db2 subsystem to Db2 13 \(Db2 Installation and Migration\)](#) or [Migrating an existing data sharing group to Db2 13 \(Db2 Installation and Migration\)](#).

4. Verify that you no longer need to fall back to Db2 12.

Important: After function level 500 is activated in Db2 13, coexistence and fallback to Db2 12 are no longer possible. You can activate function level 100* to disable new capabilities in Db2 13, but function level 100* does not support coexistence or fallback .

5. In data sharing, ensure that the group has no active Db2 12 members. See [Migrating subsequent members of a group to Db2 13 \(Db2 Installation and Migration\)](#).

Procedure

To activate function level 501, complete the following steps:

1. Generate tailored JCL jobs for the CATMAINT and function level activation steps. You can use the DSNTIJBC batch job or the Db2 installation CLIST.

Tip: You can avoid working through the Db2 installation CLIST panels in interactive mode by running a batch job with valid input files to generate the required JCL jobs and input files with a background process. See [Generating tailored Db2 migration or function level activation jobs in the background \(Db2 Installation and Migration\)](#).

To generate the required JCL jobs and input files with a background process, complete the following steps:

- a. Customize the DSNTIDOA parameter override file by following that instructions in the file.
- b. Customize the DSNTIJBC job. For example, if *prefix*.SDSNSAMP(DSNTIDOA) is the customized parameter override file, you can specify the following values in the IPSTART command in DSNTIJBC.

```
ISPSTART CMD(%DSNTINSB +  
  OVERPARM(prefix.SDSNSAMP(DSNTIDOA)) +  
  ) BREDIMAX(1)
```

- c. If you use Db2 Value Unit Edition, you must also provide the data set name of the DSNTIDVU parameter override file in the IPSTART command in the DSNTIJBC job, as shown in the following example, where *prefix*.SDSNSAMP(DSNTIDVU) is the customized OTC LICENSE file.

```
ISPSTART CMD(%DSNTINSB +  
  OVERPARM(<prefix>.SDSNSAMP(DSNTIDOA)) +  
  OTCLPARM(<prefix>.SDSNSAMP(DSNTIDVU)) +  
  ) BREDIMAX(1)
```

- d. Submit the customized DSNTIJBC job.

To generate the required JCL jobs and input files with the Db2 installation CLIST in interactive mode, complete the following steps:

- a. In panel DSNTIPA1, specify `INSTALL TYPE ===> ACTIVATE`. Then, specify the name of the output member from the previous function level activation (or migration) in the INPUT MEMBER field, and specify a new member name in the OUTPUT MEMBER field.
 - b. In panel DSNTIP00, specify the current function level and `TARGET FUNCTION LEVEL ===> V13R1M501`. The Db2 installation CLIST uses this value when it tailors the ACTIVATE command in the DSNTIJAF job and the CATMAINT utility control statement in the DSNTIJTC job.
 - c. Proceed through the remaining Db2 installation CLIST panels, and wait for the Db2 installation CLIST to tailor the jobs for the activation process. The output data set contains the tailored jobs for the activation process. For more information, see [The Db2 installation CLIST panel session \(Db2 Installation and Migration\)](#).
2. If the current function level is V13R1M100, activate function level 500 by running the generated DSNTIJA0 job, or by issuing the following ACTIVATE command:

```
-ACTIVATE FUNCTION LEVEL (V13R1M500)
```

3. Ensure that no incompatible applications can interfere with the catalog update. For more information, see Chapter 19, “Identifying applications that are incompatible with catalog updates,” on page 145.
4. Update the catalog and verify the changes for function level 501 by completing the following steps:

- a. Run the tailored DSNTIJTC job, or run the CATMAINT utility with LEVEL V13R1M501, to update the catalog to the appropriate catalog level. If multiple catalog updates are required, the CATMAINT job processes each update in sequential order. If a later update in the sequence fails, the previous successful updates do not roll back, and the catalog level remains at the highest level reached. If that occurs, you can correct the reason for the failure and resubmit the same CATMAINT job.
 - b. Run the generated DSNTIJX2 job to run the CHECK INDEX utility for Db2 catalog and directory indexes for new objects that are created in Db2 13.
5. Check that Db2 is ready for function level activation by issuing the following ACTIVATE command with the TEST option:

```
-ACTIVATE FUNCTION LEVEL (V13R1M501) TEST
```

Db2 issues message DSNU757I to indicate the results. For more information, see [Chapter 18, “Testing Db2 function level activation,”](#) on page 143.

6. Run the tailored DSNTIJAF job, or issue the following ACTIVATE command:

```
-ACTIVATE FUNCTION LEVEL (V13R1M501)
```

7. If you are ready for applications to use new capabilities in this function level, rebind them at the corresponding application compatibility level. For more information, see [Chapter 23, “Controlling the Db2 application compatibility level,”](#) on page 155.

Optionally, when you are ready for all applications to use the new capabilities of the target function level, you can run the following jobs:

- a. Run DSNTIJUZ to modify the subsystem parameter module with the APPLCOMPAT value that was specified on panel DSNTIP00.
 - b. Run DSNTIJOZ job to issue SET SYSPARM command to bring the APPLCOMPAT subsystem parameter changes online.
 - c. Run DSNTIJUA job to modify the Db2 data-only application defaults module with the SQLLEVEL value that was specified on panel DSNTIP00.

Related tasks

[Activating Db2 13 function level 500 or higher \(Db2 Installation and Migration\)](#)

[Activating Db2 13 function levels](#)

You control the activation and adoption of new features in Db2 13 by activating function levels and specifying the application compatibility level. You can also continue to apply corrective and preventative service without adopting new feature function.

Related reference

[-ACTIVATE \(Db2\) \(Db2 Commands\)](#)

Function level 500 (for migrating to Db2 13 - May 2022)

Activating function level 500 (V13R1M500) prevents coexistence with and fallback to Db2 12. Function level 500 is also the first opportunity for applications to use many of the new capabilities in Db2 13. However, new capabilities that depend on Db2 13 catalog changes remain unavailable.

Contents

[“New capabilities in function level 500” on page 72](#)

[“New-function APARS in function level 500 or higher” on page 75](#)

[“How to activate function level 500” on page 76](#)

Enabling APAR:	None
Full identifier:	V13R1M500

Catalog level required:	V13R1M100
Product identifier (PRDID):	DSN13011
Incompatible changes:	None

New capabilities in function level 500

Function level 500 activates the following new capabilities in Db2 13.

Increased flexibility for package ownership

Starting at function level 500, you can specify the type of owner for a plan, package, or service, or the type of package owner for an SQL PL routine. The owner can be a role or an authorization ID. The default owner is a role in a trusted context that is defined with the role as object owner and qualifier attributes, otherwise the default owner is an authorization ID.

For more information, see the [PACKAGE OWNER clause of CREATE PROCEDURE \(SQL - native\) \(Db2 SQL\)](#) and the [OWNERTYPE option of the OWNER bind option \(Db2 Commands\)](#).

Page sampling for inline statistics

Beginning in function level 500, the REORG TABLESPACE and LOAD utilities can now use page sampling when they gather inline statistics. Page sampling has the potential to reduce both CPU time and elapsed time. In earlier Db2 releases, the RUNSTATS utility can use page sampling, but inline statistics that are gathered by other utilities can use row sampling only. To use page sampling for inline statistics with REORG TABLESPACE or LOAD, specify the TABLESAMPLE SYSTEM option or ensure that the STATPGSAMP subsystem parameter is set to SYSTEM (the default) or YES. In function level 500, STATPGSAMP is extended to apply to inline statistics. For more information, see the [TABLESAMPLE SYSTEM option description in Syntax and options of the LOAD control statement \(Db2 Utilities\)](#) and [Syntax and options of the REORG TABLESPACE control statement \(Db2 Utilities\)](#).

SQL Data Insights

Function level 500 delivers SQL Data Insights (SQL DI), an integrated solution that brings deep learning AI capabilities into Db2. SQL DI uses unsupervised neural networks to generate a specialized vector-embedding model called database embedding, which can be referenced through SQL queries called "cognitive intelligence" queries.

The SQL DI user interface is an optional feature available at no additional charge with Db2 13, which provides the user interface for training models and exploring data insights. Db2 provides the in-database infrastructure for training and model table (vector table) management. Db2 also provides three new built-in cognitive functions to speed up query execution.

For more information, see [Running AI queries with SQL SQL Data Insights](#).

Reduced ECSA storage for IFI buffers

Db2 13 reduces the use of ECSA storage for IFI buffers from a maximum of 50 MB to a fixed 8 MB.

Function level 100 reduces the use of ECSA storage for IFI buffers to a maximum of 25 MB. Then, after function level 500 is first activated, it is further reduced to 8 MB. The storage behavior that is introduced in function level 500 continues even if you later activate function level 100*.

To compensate for the reduction in ECSA storage, you must set aside an extra 50 MB for HVCOMMON and 25 MB for private storage. You can reduce the ECSA storage after function level 500 is activated and Db2 starts using the new storage pools. When Db2 uses the new storage pools, the use of ECSA for the retrieval of IFI records noticeably decreases. You can monitor use of the new storage pools by starting the statistics trace to collect IFCID 225. Then, you can check the SHARED / COMMON storage summary report in the formatted IDCID 225 SMF trace record.

For more information about ECSA storage requirements, see [Calculating the storage requirement for the extended common service area \(Db2 Installation and Migration\)](#).

Online conversion of tables from growth-based (PBG) to range-based (PBR) partitions

Function level 500 introduces the capability to convert the partitioning scheme of a table with growth-based partitions (in a PBG table space) to use range-based partitions (in a PBR table space). The conversion can be completed as an online change with minimal impact to your applications.

PBG and PBR universal table spaces (UTS) are the strategic table space types for tables in Db2 for z/OS. PBG table spaces are the default UTS type, and they are well-suited for small to medium-sized tables. However, if an existing table in a PBG table space grows too large, performance degradation or data and index management issues might arise. Consider converting from PBG to PBR when that occurs.

To complete the conversion, you issue an ALTER TABLE statement with the new ALTER PARTITIONING TO PARTITION BY RANGE clause and run the REORG TABLESPACE utility to materialize the pending change. The table space for the table is converted to PBR with relative page numbering (RPN).

For more information, see [Converting tables from growth-based to range-based partitions \(Db2 Administration Guide\)](#) and "ALTER PARTITIONING TO PARTITION BY RANGE" in [ALTER TABLE \(Db2 SQL\)](#).

Fast index traversal (FTB) support for larger index keys

Function level 500 extends FTB support to unique indexes with a key size for the ordering columns up to 128 bytes and nonunique indexes with a key size up to 120 bytes. For more information, see [Fast index traversal \(Db2 Performance\)](#).

Increased control for applications over how long to wait for a lock

Function level 500 introduces the CURRENT LOCK TIMEOUT special register and the SET CURRENT LOCK TIMEOUT SQL statement to allow the lock timeout value to be set at the application level. So, you can set a lock timeout interval that suits the needs of a specific application, or even an individual SQL statement. Doing so minimizes application lock contention and simplifies portability of applications to Db2, without the need to assign the application to a separate Db2 subsystem.

The value of the CURRENT LOCK TIMEOUT special register overrides the value of the IRLMRWT subsystem parameter. It applies to certain processes related to locking, like the claim or drain of an object and cached dynamic statement quiescing.

For more information, see [CURRENT LOCK TIMEOUT special register \(Db2 SQL\)](#) and [SET CURRENT LOCK TIMEOUT \(Db2 SQL\)](#).

You can limit use of CURRENT LOCK TIMEOUT by setting the new SPREG_LOCK_TIMEOUT_MAX subsystem parameter. For more information, see [LOCK TIMEOUT MAX \(SPREG_LOCK_TIMEOUT_MAX subsystem parameter\) \(Db2 Installation and Migration\)](#).

You can also use Db2 profile tables to specify an assignment for the CURRENT LOCK TIMEOUT special register, for both remote and local threads. See [Setting special registers by using profile tables \(Db2 Administration Guide\)](#).

Profile table enhancements for application environment settings

Db2 13 introduces the capability to use system profiles for local applications in certain situations. Previously, the initial values for special registers and system built-in global variables can be specified in the Db2 profile tables, but they are used only for initialization with distributed threads. The new Db2 profile table support for local applications requires Db2 to be started with the DDF subsystem parameter set to AUTO or COMMAND. See [DDF STARTUP OPTION field \(DDF subsystem parameter\) \(Db2 Installation and Migration\)](#).

Starting in function level 500, Db2 profile tables can now be used for both local and remote applications in the following situations:

- You can specify assignments to the new CURRENT LOCK TIMEOUT special register. For more information, see [CURRENT LOCK TIMEOUT special register \(Db2 SQL\)](#) and [Setting special registers by using profile tables \(Db2 Administration Guide\)](#).

- You can specify a new `RELEASE_PACKAGE` keyword with a `COMMIT` attribute to change the release bind option for a package. See [Overriding the `RELEASE\(DEALLOCATE\)` option for packages by using profile tables \(Db2 Performance\)](#).

Ability to delete an active log data set from the BSDS while Db2 is running

Function level 500 introduces the new `REMOVELOG` option for the `-SET LOG` command to support online removal of an active log data set from the BSDS, eliminating the need to stop Db2 to accomplish the task by using the offline utility `DSNJU003`. The `-SET LOG REMOVELOG` command deletes the specified log from the BSDS if it is not in use or mark the log `REMOVAL PENDING` if it is in use.

To provide monitoring of the current active log status for log data sets with `REMOVAL PENDING` status, function level 500 also introduces the `DETAIL` option for the `-DISPLAY LOG` command. It shows information regarding `REMOVAL PENDING` status for local active log data sets. The output from the utility `DSNJU004` also shows the `REMOVAL PENDING` status where applicable.

For more information, see [Deleting an active log data set from the BSDS with the `-SET LOG` command \(Db2 Administration Guide\)](#).

SPT01 and SYSLGRNX table spaces are converted to DSSIZE 256 GB

Starting in function level 500, the first time that the `REORG TABLESPACE` utility runs for the following directory objects, it converts the `DSSIZE` to 256 GB.

- `DSNDB01.SPT01` to resolve issues that are related to the removal of the `SPT01_INLINE_LENGTH` subsystem parameter by APAR PH24358 in Db2 12.
- `DSNDB01.SYSLGRNX` in anticipation of future growth in this table for increasing workloads and conversions of non-UTS table space to UTS.

The conversion is automatic and does not require any special utility syntax. It updates the following Db2 catalog table values for each table space:

- The `DSSIZE` columns in `SYSIBM.SYSTABLESPACE` and `SYSIBM.SYSTABLEPART` are updated to 256G.
- A `SYSCOPY` record is inserted for the table space, with the following values to indicate that `REORG` changed the `DSSIZE`: `ICTYPE = 'A'`, `STYPE = 'D'`, `TTYE = '64G'`.

If function level 100* is activated, already converted table spaces continue to use the larger `DSSIZE`, but the `REORG` utility does not convert unconverted table spaces.

Recovery to a point-in-time (PIT) before `REORG` converted the `DSSIZE` reverts the `DSSIZE` to 64GB. As always, if any one of the catalog or directory objects are recovered to a prior PIT, it is best to recover all catalog and directory objects to the same PIT.

Improved concurrency for altering tables for DATA CAPTURE

Function level 500 introduces a concurrency improvement for `ALTER TABLE` statements that change the `DATA CAPTURE` attribute of tables. With this enhancement, Db2 no longer waits for other statements that depend on the altered table to commit. As a result, the `DATA CAPTURE` alteration can now succeed even when concurrent statements are running continually against the table.

Earlier Db2 releases quiesce the following objects that depend on the altered table as part of the `DATA CAPTURE` alteration:

- Static packages
- Cached dynamic SQL statements

Because the `DATA CAPTURE` alteration waited for applications that depended on the altered table to commit, continuous concurrent activity on the table might cause the `ALTER TABLE` statements to fail.

The new `DATA CAPTURE` attribute now takes effect immediately when the processing completes, even before the `ALTER` statement commits. As a result, concurrent statements on the same Db2 member might write out different log formats in the same transaction. For more information, see [Altering a table to capture changed data \(Db2 Administration Guide\)](#).

Change REORG INDEX SHRLEVEL REFERENCE or CHANGE so the NOSYSUT1 keyword is the default

Starting at function level 500, the NOSYSUT1 keyword is the default for the REORG INDEX utility when specified with the SHRLEVEL REFERENCE or CHANGE keywords. So, the utility avoids use of a work data set, which improves performance and allows REORG INDEX to use parallel subtasks to unload and to build index keys. The default value of the REORG_INDEX_NOSYSUT1 subsystem parameter is also changed from NO to YES, and YES is now the only option. So, this subsystem parameter no longer influences the behavior of REORG INDEX.

For more information, see [Syntax and options of the REORG INDEX control statement \(Db2 Utilities\)](#) and [REORG TS NOPAD DEFAULT \(REORG_TS_NOPAD_DEFAULT subsystem parameter\) \(Db2 Installation and Migration\)](#).

Long names support for timeout and deadlock messages in IRLM

Starting at function level 500, Db2 13 introduces IRLM support for long names for client information such as workstation ID, user ID, and transaction ID, in deadlock and timeout messages DSNT175I and DSNT376I.

In Db2 12 and earlier, the long name values are truncated in the message output.

With this change, Db2 13 also starts populating existing long name fields in IFICID 172 and IFCID 196 records. These fields remain unpopulated in Db2 12 and earlier.

CREATE TABLESPACE uses MAXPARTITIONS 254 by default

At application compatibility level V13R1M500 or higher, CREATE TABLESPACE statements use MAXPARTITIONS 254 by default.

When MAXPARTITIONS 256 is explicitly specified, the default DSSIZE varies from 4 G to 32 G depending on the page size. However, starting with application compatibility level V12R1M504, when MAXPARTITIONS is not explicitly specified, Db2 12 use MAXPARTITIONS 256 by default, but the default DSSIZE is always 4 G regardless of the page size.

This apparent inconsistency avoided a risk of failure for existing statements, where the default data set size might be greater than 4 G depending on the page size. The statements might fail with SQLCODE -904 with reason code 00D70008 if the data sets for the table space are not associated with a DFSMS data class that is specified with extended format and extended addressability.

With MAXPARTITIONS 254 as the default, the result is now consistent regardless of whether MAXPARTITIONS is explicitly specified. The calculated default DSSIZE is always 4 G.

See the MAXPARTITIONS and DSSIZE descriptions in [CREATE TABLESPACE \(Db2 SQL\)](#).

New-function APARS in function level 500 or higher

The following capabilities that are introduced by new-function APARs after the general availability of Db2 13 take effect when you activate function level 500 with the PTF applied, or immediately if you apply the PTF at any higher function level.

Improved DBAT status for MONITOR THREADS profiles in DISPLAY THREAD output

FL 500 Starting in Db2 13 at function level 500 or higher, improves the status values in DISPLAY THREAD output for DBATs that are queued because the MAXDBAT subsystem parameter or the except threshold for a MONITOR THREADS profile was reached. This APAR introduces the new status value RS in DSNV402I to indicate a thread that is suspended because a MONITOR THREADS profile was reached. The existing RQ value is also updated to indicate that it only applies that threads that are suspended because the MAXDBAT value was reached.

This APAR also adds a new counter in the output for the DISPLAY DDF command with the DETAIL option. The new PQDBAT counter in message DSNL093I indicates the current number of DBATs queued because a system profile exception threshold was reached.

FL 504 Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [ST in DSNV401I](#)
- [PQDBAT in DSNL080I](#)
- [PH47626](#)

Reduced LOGREC entries for parallel queries with FETCH FIRST *n* ROWS

Starting in Db2 13 function level 500 or higher, APAR PH48183 (September 2022) reduces the number of EREP LOGREC entries with reason code 00E50013 for parallel queries that specify FETCH FIRST *n* ROWS.

If a query that specifies FETCH FIRST *n* ROWS runs in parallel, the parallel child tasks continue fetching more records as the parent retrieves and returns them to the application. As soon as *n* rows are returned, the parent sends a "stop" message to the child tasks. When each child receives the stop message, it stops processing as if canceled. Before this APAR, each child task also writes out an EREP LOGREC entry with reason code 00E50013 in this situation.

With APAR PH48183 applied, the parallel child tasks can stop processing sooner and avoid issuing the EREP LOGREC entries with 00E50013.

The new logic applies for applications that run at application compatibility V13R1M500 or higher.

FL 504 Activation of function level 504 or higher verifies that the PTF for this APAR is applied on every data sharing member, but its changes take effect at any function level.

For more information, see the following related topics:

- [00E50013 \(Db2 Codes\)](#)
- [PH48183](#)

V13R1M500 application compatibility

Most new SQL capabilities become available only to applications that use the equivalent application compatibility (APPLCOMPAT) level or higher. For a list, see [Chapter 7, "SQL changes in Db2 13," on page 99](#).

For more information about application compatibility levels, see [Chapter 23, "Controlling the Db2 application compatibility level," on page 155](#).

How to activate function level 500

Before you begin

Before you activate function level 500 or higher, complete the following prerequisite tasks:

1. In Db2 12, identify and resolve incompatible changes and activate function level 510 (V12R1M510). You can run the pre-migration job DSNTIJPE in Db2 12 to identify the incompatible changes. For more information, see [Verify Db2 13 premigration activities and activate function level 510 in Db2 12 \(Db2 Installation and Migration\)](#).

2. Verify that every member was restarted with the fallback SPE applied in Db2 12.

Important: Inactive members that never started in Db2 12 with the fallback SPE (APAR PH37108) applied cannot start after the first data sharing member is migrated to Db2 13 at function level 100.

3. Migrate the Db2 subsystem or data sharing group to Db2 13, as described in [Migrating your Db2 subsystem to Db2 13 \(Db2 Installation and Migration\)](#) or [Migrating an existing data sharing group to Db2 13 \(Db2 Installation and Migration\)](#).

4. Verify that you no longer need to fall back to Db2 12.

Important: After function level 500 is activated in Db2 13, coexistence and fallback to Db2 12 are no longer possible. You can activate function level 100* to disable new capabilities in Db2 13, but function level 100* does not support coexistence or fallback.

5. In data sharing, ensure that the group has no active Db2 12 members. See [Migrating subsequent members of a group to Db2 13 \(Db2 Installation and Migration\)](#).

Procedure

To activate function level 500, complete the following steps:

1. Generate tailored JCL jobs for the CATMAINT and function level activation steps. You can use the DSNTIJBC batch job or the Db2 installation CLIST.

Tip: You can avoid working through the Db2 installation CLIST panels in interactive mode by running a batch job with valid input files to generate the required JCL jobs and input files with a background process. See [Generating tailored Db2 migration or function level activation jobs in the background \(Db2 Installation and Migration\)](#).

To generate the required JCL jobs and input files with a background process, complete the following steps:

- a. Customize the DSNTIDOA parameter override file by following that instructions in the file.
- b. Customize the DSNTIJBC job. For example, if *prefix*.SDSNSAMP(DSNTIDOA) is the customized parameter override file, you can specify the following values in the IPSTART command in DSNTIJBC.

```
ISPSTART CMD(%DSNTINSB +  
  OVERPARM(prefix.SDSNSAMP(DSNTIDOA)) +  
  ) BREDIMAX(1)
```

- c. If you use Db2 Value Unit Edition, you must also provide the data set name of the DSNTIDVU parameter override file in the IPSTART command in the DSNTIJBC job, as shown in the following example, where *prefix*.SDSNSAMP(DSNTIDVU) is the customized OTC LICENSE file.

```
ISPSTART CMD(%DSNTINSB +  
  OVERPARM(<prefix>.SDSNSAMP(DSNTIDOA)) +  
  OTCLPARM(<prefix>.SDSNSAMP(DSNTIDVU)) +  
  ) BREDIMAX(1)
```

- d. Submit the customized DSNTIJBC job.

To generate the required JCL jobs and input files with the Db2 installation CLIST in interactive mode, complete the following steps:

- a. In panel DSNTIPA1, specify `INSTALL TYPE ==> ACTIVATE`. Then, specify the name of the output member from the previous function level activation (or migration) in the INPUT MEMBER field, and specify a new member name in the OUTPUT MEMBER field.
 - b. In panel DSNTIP00, specify `TARGET FUNCTION LEVEL ==> V13R1M500`. The Db2 installation CLIST uses this value when it tailors the ACTIVATE command in the DSNTIJAF job and the CATMAINT utility control statement in the DSNTIJTC job. (Function level 500 uses catalog level 100, and the tailored DSNTIJTC job is not used.)
 - c. Proceed through the remaining Db2 installation CLIST panels, and wait for the Db2 installation CLIST to tailor the jobs for the activation process. The output data set contains the tailored jobs for the activation process. For more information, see [The Db2 installation CLIST panel session \(Db2 Installation and Migration\)](#).
2. Check that Db2 is ready for function level activation by issuing the following ACTIVATE command with the TEST option:

```
-ACTIVATE FUNCTION LEVEL (V13R1M500) TEST
```

Db2 issues message DSNU757I to indicate the results. For more information, see [Chapter 18, “Testing Db2 function level activation,” on page 143](#).

3. Run the tailored DSNTIJAF job, or issue the following ACTIVATE command:

```
-ACTIVATE FUNCTION LEVEL (V13R1M500)
```

4. If you are ready for applications to use new capabilities in this function level, rebind them at the corresponding application compatibility level. For more information, see [Chapter 23, “Controlling the Db2 application compatibility level,” on page 155](#).

Optionally, when you are ready for all applications to use the new capabilities of the target function level, you can run the following jobs:

- a. Run DSNTIJUZ to modify the subsystem parameter module with the APPLCOMPAT value that was specified on panel DSNTIP00.
- b. Run DSNTIJOZ job to issue SET SYSPARM command to bring the APPLCOMPAT subsystem parameter changes online.
- c. Run DSNTIJUA job to modify the Db2 data-only application defaults module with the SQLLEVEL value that was specified on panel DSNTIP00.

What to do next

To activate new capabilities with catalog dependencies in Db2 13, activate function level 501 or higher. See [“Function level 501 \(Db2 13 installation or migration - May 2022\)” on page 67](#).

Related tasks

[Activating Db2 13 function level 500 or higher \(Db2 Installation and Migration\)](#)

Related reference

[-ACTIVATE \(Db2\) \(Db2 Commands\)](#)

Function level 100 (for migrating to Db2 13 - May 2022)

Db2 starts at function level 100 (V13R1M100) during migration to Db2 13, and fallback and coexistence with Db2 12 in data sharing remain possible. Many new capabilities in Db2 13 remain unavailable.

Contents

[“New capabilities in function level 100” on page 78](#)

[“How to migrate to Db2 13 function level 100” on page 86](#)

Enabling APAR:	None
Full identifier:	V13R1M100
Catalog level required:	V13R1M100
Product identifier (PRDID):	DSN13010
Incompatible changes:	See Chapter 4, “Incompatible changes in Db2 13,” on page 89 .

New capabilities in function level 100

In function level 100 (V13R1M100), Db2 runs on Db2 13 code, and virtual storage and many optimization enhancements in Db2 13 become available. However, most new application and SQL capabilities remain unavailable until you activate the function level that introduces them.

The following capabilities and enhancements in Db2 13 become available in function level 100.

Index look-aside optimization

Function level 100 introduces index look-aside optimizations, to improve performance for insert, update, and delete operations. This enhancement optimizes Db2 13 and improves its performance without changing how you configure, monitor, and use Db2.

Sort optimization

Function level 100 introduces enhanced sort optimizations, which were previously introduced for ORDER BY and GROUP BY processing. It applies them to improve the performance of certain operations, such as the following processing enhancements:

- Machine-generated code support for DECFLOAT processing.
- Support for the following enhancements for GROUPING SET, multiple DISTINCT, and PERCENTILE processing:

- Machine-generated code support.
- Sort processing can use its own work file.
- A check for ordered data in the first iteration of a sort.
- Larger sort trees can be used.
- SUBSTR support for the LISTAGG built-in function, if the start position and length for SUBSTR is a constant.
- Support for avoiding rereading of a single work file, if the sort work file is used, and if IBM Watson Machine Learning for z/OS is enabled.
- Support for reducing the length of long VARCHAR keys, if the last key in an ORDER BY list is a VARCHAR over 100 bytes, and if IBM Watson Machine Learning for z/OS is enabled.

This enhancement optimizes Db2 13 and improves its performance without changing how you configure, monitor, and use Db2.

Expanded SORTL usage with learning from execution (IBM z15)

Function level 100 introduces expanded SORTL usage based on machine learning on the amount of storage and the number of records being sorted, when run on IBM z15 or later processors. This enhancement optimizes Db2 13 and improves its performance without changing how you configure, monitor, and use Db2.

Improved locking for INSERT to partition-by-growth (PBG) table spaces

Function level 100 introduces retry logic for INSERT operations. An extra attempt is made to obtain a partition lock on a PBG table space after a failed first attempt, thereby increasing the success rate of INSERT operations.

Before this enhancement, only a single attempt was made to obtain a lock on the target partition. If the attempt failed, the target partition was skipped, and the next partition was evaluated. This process would continue until the INSERT operation either successfully obtained a partition lock or it finished searching all existing partitions without obtaining a partition lock.

In most cases, the duration of partition lock contention is short; however, because the INSERT operation did not make another attempt to obtain a lock on a partition after the first failed attempt, the INSERT operation terminated unnecessarily. In many cases, making an extra attempt to obtain a partition lock results in the successful completion of an INSERT operation that otherwise would fail.

This enhancement optimizes Db2 13 and improves its performance without changing how you configure, monitor, and use Db2.

Improved cross-partition search for INSERT to PBG table spaces

Starting in function level 100, if free space is not found in the initial target partition for an insert to a partition-by-growth (PBG) table space, Db2 13 more efficiently uses trailing empty partitions other than the last physical partitions.

In Db2 12 and later, the cross-partition search for PBG table spaces is bidirectional. That is, when an insert operation fails to find free space in the initial target partition, which is selected based on the clustering index, the partition to be searched next can be either an ascending or descending partition sequence number. The searching order is randomly decided at run time to avoid creating a “hot spot” in a single partition. When Db2 12 reaches the first physical partition during a descending partition search, it wraps around and looks at the last physical partition next. As a result, when a PBG table space has many empty partitions at the end, the descending cross-partition search algorithm in Db2 12 often uses the last physical partitions, and it can sometimes leave many unused empty partitions unused in between the first and last partitions.

To prevent this situation, Db2 13 now tracks the highest non-empty partition in the table space in real-time statistics at run time. The tracking starts when a data set is opened and continues until it is closed. When Db2 13 uses a descending cross-partition search and after it reaches the first partition, Db2 13 now searches the cached highest non-empty partition next, instead of the last physical partitions.

This enhancement optimizes Db2 13 and improves its performance without changing how you configure, monitor, and use Db2.

Reduced ECSA storage for IFI buffers

Db2 13 reduces the use of ECSA storage for IFI buffers from a maximum of 50 MB to a fixed 8 MB.

Function level 100 reduces the use of ECSA storage for IFI buffers to a maximum of 25 MB. Then, after function level 500 is first activated, it is further reduced to 8 MB. The storage behavior that is introduced in function level 500 continues even if you later activate function level 100*.

To compensate for the reduction in ECSA storage, you must set aside an extra 50 MB for HVCOMMON and 25 MB for private storage. You can reduce the ECSA storage after function level 500 is activated and Db2 starts using the new storage pools. When Db2 uses the new storage pools, the use of ECSA for the retrieval of IFI records noticeably decreases. You can monitor use of the new storage pools by starting the statistics trace to collect IFCID 225. Then, you can check the SHARED / COMMON storage summary report in the formatted IDCID 225 SMF trace record.

For more information about ECSA storage requirements, see [Calculating the storage requirement for the extended common service area \(Db2 Installation and Migration\)](#).

Reduced agent local below-the-bar (BTB) storage

Starting in function level 100, Db2 supports a greater number of concurrent threads, by using above-the-bar (ATB) agent-local storage for statement text and attribute strings for dynamic SQL statements. In earlier releases, Db2 kept a copy of dynamic SQL statement text and attribute strings in agent local below-the-bar (BTB) storage while the statement is being prepared and executed.

For any specific thread, multiple dynamic SQL statements can be executing depending on the nesting level. The maximum length of an SQL statement is 2 MB, but much more storage can be allocated and the consumption of BTB storage could prevent the number of threads from scaling.

This enhancement optimizes Db2 13 and improves its performance without changing how you configure, monitor, and use Db2.

DBAT availability improvements

Function level 100 introduces changes to Db2 13 DBAT termination processing to support the following objectives:

- Reduction of the overall frequency and number of DBAT terminations.
- Reduction of the number of concurrent DBAT terminations that are caused by a short-term increase in DBAT usage.

This enhancement optimizes Db2 13 and improves its performance without changing how you configure, monitor, and use Db2.

Improved performance when using external security

Function level 100 introduces the following enhancements to improve performance for Db2 environments that use external security:

- Db2 caches plan authorization checks that the access control authorization exit (ACAE) routine uses. Previously, successful authorization checks on the EXECUTE privilege for plans were not cached if those checks were completed by the ACAE routine. This enhancement provides consistent behavior in plan authorization cache behavior regardless of whether security is managed with Db2 facilities or with the ACAE.

To enable plan authorization caching when the ACAE routine is being used, the AUTHEXIT_CACHEREFRESH subsystem parameter must be set to ALL and the z/OS release must be 2.5 or later. Db2 caches the results of authorization checks on the EXECUTE privilege for plans if a profile in the RACF class MDSNPN permits access to the plan. Db2 does not cache the results if access is allowed due to administrative authority, such as the DATAACCESS or SYSADM authorities.

For more information, see [Caching of EXECUTE on plans, packages, and routines \(Managing Security\)](#).

- Db2 is enhanced to cache more authorization IDs per plan. For more information, see [Caching authorization IDs for plans \(Managing Security\)](#).
- The AUTHCACH subsystem parameter is removed to simplify plan authorization cache management. Use the CACHESIZE bind option on the BIND PLAN subcommand to specify the size of the authorization cache for that plan. The default value is 4K.
- If the AUTHEXIT_CACHEREFRESH subsystem parameter is set to ALL, the global authentication cache takes the timestamp into consideration for user IDs that were authenticated by using credentials other than multi-factor authentication (MFA). For more information, see [Global authentication cache \(Managing Security\)](#).
- When you specify a key label for data set encryption, the specified key label cannot refer to an archived key for decryption operations only. Key labels can be specified by using the ENCRYPTION_KEYLABEL subsystem parameter or any of the following SQL statements:
 - ALTER STOGROUP
 - ALTER TABLE
 - CREATE STOGROUP
 - CREATE TABLE

If the specified key label refers to a decryption-only archived key, the key label specification fails and returns an error message. For more information on decryption-only archived keys, see [ICSF: Limit archived keys to decrypt operations only](#).

Reduced ECSA storage use for distributed data facility (DDF) processing

Function level 100 reduces the amount of ECSA storage that is used for processing DDF threads to be equivalent to processing local threads. The previous recommendation was an extra 2 KB per DDF thread. For more information, see [Calculating the storage requirement for the extended common service area \(Db2 Installation and Migration\)](#).

Improved storage monitoring and contraction

Function level 100 introduces the following enhancements to provide storage constraint relief:

- When below-the-bar Db2 storage consumption exceeds 64-percent threshold, Db2 automatically begins contraction of private storage pools.
- When extended common service area (ECSA) storage consumption exceeds the 85-percent threshold, Db2 automatically begins contraction of storage pools that are allocated in the ECSA.

In both cases, the storage contraction stops after storage consumption drops below the threshold.

This enhancement optimizes Db2 13 and improves its performance without changing how you configure, monitor, and use Db2.

Dynamic management of CF lock storage by IRLM

With IRLM 2.3 at function level 50C or higher, which is included with Db2 13, IRLM can now invoke an existing capability in z/OS Sysplex Services for Data sharing (XES) to dynamically expand the coupling facility (CF) lock structure storage size. This new internal monitoring capability in IRLM can improve lock request processing throughput, by expanding the CF lock structure size to process lock requests, instead of rejecting them.

The existing XES monitoring of the CF lock structure use is defined in the coupling facility resource management (CFRM) policy as a threshold percentage value, and it is enabled with a default of 80% when it is not equal to zero. This monitoring retrieves statistics on the CF lock structure every 60 seconds when the storage usage is less than the threshold and every 30 seconds when the storage usage is equal to or greater than the full threshold. IRLM can determine the storage needed at a higher level of granularity than the existing monitoring by XES of CF structure, especially when a spike in locking activities results in rejection of lock requests due to insufficient Record List Entries (RLEs) even before z/OS has a chance to start the CF lock structure alteration.

The existing structure monitoring by XES handles storage contraction, and it contracts all eligible structures in the coupling facility by 10 percent in each cycle when the entire coupling facility is at or more than 90% full.

IRLM issues the following messages when it adjusts the CF lock structure storage size: [DXR189I](#) and [DXR190I](#).

Improved Db2 installation and migration process for customizing the amount of private storage for IRLM locks

In the MAX STORAGE FOR LOCKS field on installation panel DSNTIPJ, you can specify the maximum amount of private storage above the 2 GB bar for the IRLM lock control structure. In earlier Db2 releases, you can specify a value of only up to 102400 megabytes. Starting in Db2 13, you can specify a value of up to 16384 petabytes.

For more information, see MAX STORAGE FOR LOCKS field ([Db2 Installation and Migration](#)) and MAX LOCK STORAGE UNIT field ([Db2 Installation and Migration](#)).

Memory usage reduced for REBIND with APREUSE

Function level 100 introduces reduced storage usage during BIND/REBIND for queries that involve many tables. This enhancement optimizes Db2 13 and improves its performance without changing how you configure, monitor, and use Db2.

Partition range support in IFCID 306 for users of replication applications

Function level 100 introduces the capability for applications that collect IFCID 306 trace records for the log read process to request filtering on a range of partitions. For more information, see the new WQLSFLG flag and WQLSDBPP mapping in [Qualification fields for READS requests \(Db2 Performance\)](#).

EDITPROC support in IFCID 306 for users of nonproxy mode replication applications

Function level 100 introduces the support for any user of IFCID 306 in non-proxy mode to use EDITPROC support as an on-request function. For more information, see the new values X'04', X'05', X'06', and X'07' for the 1-byte WQALLOPT field in [Qualification fields for READS requests \(Db2 Performance\)](#).

Relative page numbering for new PBR table spaces

Starting in function level 100, the default value of the PAGESET_PAGENUM subsystem parameter is changed to RELATIVE. The PAGESET_PAGENUM subsystem parameter specifies the default value that Db2 uses when you omit the PAGENUM option in CREATE TABLESPACE or CREATE TABLE statement for a partition-by-range (PBR) table space. That is, it specifies whether Db2 creates the table space and associated partitioned indexes to use relative page numbers (RPN) or absolute page numbers (APN) across partitions. RPN is the strategic direction for PBR table spaces in Db2. If you accept the new default and create all new PBR table spaces with relative page numbers, you can avoid costly future conversions. Converting from absolute to relative pages numbers always requires a REORG of the entire table space.

See PAGE SET PAGE NUMBERING field (PAGESET_PAGENUM subsystem parameter) ([Db2 Installation and Migration](#)).

Improved default statistics collection granularity

Starting in function level 100, that default value of the STATIME_MAIN subsystem parameter is changed to 10 seconds. The STATIME_MAIN subsystem parameter specifies the time interval in seconds, for collection of interval-driven statistics that are not collected at the interval that is specified by the STATIME subsystem parameter. With the default statistics interval set at 60 seconds in earlier releases, Db2 database administrators and system programmers cannot identify true workload peaks by using Db2 statistics for subsystem level performance tuning and planning. Similarly, a slowdown of 5 - 15 seconds is difficult to diagnose with statistics collected at a 60-second interval.

See MAIN STATS TIME field (STATIME_MAIN subsystem parameter) ([Db2 Installation and Migration](#))

REPAIR utility WRITELOG for decompression dictionaries

Function level 100 introduces the capability to write a decompression dictionary log record up to the maximum log record size supported by Db2. This capability can be used after you run an application or utility that builds a new dictionary without writing the old one to the log. So, it is useful for replication products that would otherwise require a refresh of the replication target. After the decompression dictionary is successfully written, the REPAIR utility issues message DSNU3335I with

the location of the log record. Applications can then use this information to insert a SYSIBM.SYSCOPY record.

For more information, see the option descriptions for TYPE X'4002 and SUBTYPE 'X'000A' in [Syntax and options of the REPAIR control statement \(Db2 Utilities\)](#), and [DSNU3335I \(Db2 Messages\)](#).

Enhanced space-level recovery with the RECOVER utility

Starting in function level 100, the RECOVER utility supports space-level recovery (where DSNUM ALL is specified or is the default), even if the image copies were created at the partition or piece level for the following objects:

- Universal table spaces (UTS).
- Index spaces or indexes for a UTS.
- XML UTS with a base table that resides in a UTS.
- Auxiliary index spaces or indexes for an XML UTS with a base table that resides in a UTS.
- LOB table spaces with a base table that resides in a UTS.
- Auxiliary index spaces or indexes for LOB table spaces with a base table that resides in a UTS.

Space-level recovery of these supported object types is processed as partition-level or piece-level recoveries in the RESTORE phase. Db2 catalog and directory objects that meet the criteria above are supported by this enhancement. When the list contains a mixture of supported and unsupported object types, recovery behavior for the unsupported object types is the same as Db2 12 or earlier releases.

When the RECOVER utility is invoked (with DSNUM ALL specified or as the default) in Db2 12 or earlier releases, for space-level recovery of table spaces, index spaces, or indexes, an error message is issued if the image copies were created at the partition or piece level. The DSNU512I (DATASET LEVEL RECOVERY IS REQUIRED) error message indicates that recovery cannot be done at the space level. Recovery must instead be requested at the partition or piece level. Objects with these errors are not recovered and the RECOVER ends with RC8 indicating errors were encountered.

For more information, see [Recovering a data set or partition \(Db2 Utilities\)](#).

Column names longer than 30 bytes

Function level 100 extends the maximum length of a column name from 30 bytes of EBCDIC, up to 128 bytes with limited support for using the longer column names. The longer column names can be used when the TABLE_COL_NAME_EXPANSION subsystem parameter setting is ON. Although you can now define a column with a name up to 128 bytes, column names with a length greater than 30 bytes of EBCDIC might be truncated on a character boundary. Column names returned in an SQLDA contain 30 bytes at most. APIs that do not use the SQLDA to obtain a column name might return complete column names.

For more information, see [Column names longer than 30 bytes \(Db2 SQL\)](#) and [TABLE_COL_NAME_EXPANSION in macro DSN6SPRM \(Db2 Installation and Migration\)](#).

Db2 support for z/OS continuous compliance

Customers are looking for solutions that provide evidence that they can trust the security of z/OS systems. z/OS 2.5 introduces new SMF type 1154 records that provide evidence of security compliance. Participating components and products can collect and write compliance data to their associated SMF 1154 subtype records. Function level 100 adds the capability to collect evidence on Db2 subsystems' compliance by writing SMF 1154 subtype 81 records. For more information, see [Db2 evidence for z/OS continuous compliance \(Managing Security\)](#) and [What is new in z/OS \(V2R4 - V2R5\)](#).

More concurrent open data sets in z/OS 2.5

In function level 100 and z/OS 2.5 or later, dynamic allocation processing supports system work blocks (SWBs) for data sets that are in 64-bit storage. This new dynamic allocation function helps reduce below-the-bar storage usage for address spaces that allocate large numbers of data sets.

To enable this feature, complete one of the following actions:

- Update the ALLOCxx parmlib member to set the SYSTEM SWBSTORAGE value to ATB. The default value is SWA, which indicates that SWBs reside in 24-bit storage or 31-bit storage. ATB indicates that SWBs are allowed to reside in 64-bit storage.
- Issue system command SETALLOC SYSTEM,SWBSTORAGE=ATB.

It is best to update the ALLOCxx parmlib member because the change remains effective across IPLs. If the SETALLOC command is used to enable SYSTEM SWBSTORAGE, you must restart Db2 for the change to take effect.

Previously, the CLIST calculation for data set storage size used 5 KB per open data set. With the new dynamic allocation function, the storage that is required per open data set is reduced to 4 KB. You must adjust the calculation for data set storage size. For more information, see [Calculating data set control block storage \(Db2 Installation and Migration\)](#).

More efficient cleanup for above-the-bar storage

Function level 100 introduces improvements to how Db2 manages and frees above-the-bar storage, especially to reduce the disruptive impact of issuing excessive IARV64 REQUEST(DISCARDDATA) service requests.

Db2 13 no longer issues the IARV64 REQUEST(DISCARDDATA) request during thread deallocation or at certain intervals of COMMIT, and enhanced storage management is no longer controlled by the REALSTORAGE_MANAGEMENT subsystem parameter, which is also removed. In Db2 13, the storage is returned to the memory object. A system-level timer drives contraction for the memory object to release unused frames back to z/OS. Also, Db2 13 periodically checks the available free frames before the LPAR starts to page (by using the z/OS calculations for available free frames and LO threshold). If this value becomes lower than 5 times the z/OS calculated OK threshold, the memory object contraction is triggered.

Index page split instrumentation enhancements

Function level 100 introduces IFCID 396 to provide detailed information about index splits. When data is inserted into a base table, the corresponding indexes are modified accordingly. As a result, performance of SQL insert operations can be impacted during the *index split* process, where synchronous I/O is required under data sharing and the group buffer pool GBP is dependent on the related index page sets.

The existing IFCID 359 records already contain information index split events. However, the information that is recorded is not detailed enough to identify the cause of performance issues. IFCID 359 is also disabled by default, and it can miss capturing some abnormal index split situations.

396 is always enabled by default under statistics trace class 3 and performance trace class 6. An IFCID 396 record is generated when an index split is considered an abnormal split process, such as when the total elapsed time is greater than 1 second. The generated IFCID 396 record contains information such as the DBID, PSID, member ID, URID, page number, and more. The information is helpful for both customers and IBM Support to identify the root cause of INSERT performance issues.

For more information, see the IFCID 396 descriptions in [Chapter 11, “IFCID changes in Db2 13,”](#) on [page 111](#) and in the [Trace field descriptions \(Db2 Performance\)](#).

Starting after the catalog level V13R1M501 update (which requires function level 500), the following RTS columns in the SYSIBM.SYSINDEXSPACESTATS catalog table are populated. They record and aggregate general index split information since last table reorganization, index rebuild, or load replace:

Column name	Data type	Description
REORGTOTALSPLITS	INTEGER	The number of index splits since last reorganization or rebuild.
REORGSPPLITIME	BIGINT	Aggregated elapsed time for all index splits since last reorganization or rebuild.

Column name	Data type	Description
REORGEXCSPLITS	INTEGER	The number of abnormal index splits (such as elapsed times greater than 1 second) since last reorganization or rebuild.

Db2 starts populating these RTS columns as soon as the catalog level V13R1M501 update completes, even before function level 501 is activated.

Accounting information on the longest wait times for common suspension types

When Db2 transactions take a long time, it is important to determine:

- Where the transaction time is spent.
- Whether the problem is many short suspensions or a few long suspensions.
- Which resources the suspensions are for.

Before this enhancement, detailed performance traces were required to find this information. This enhancement simplifies the diagnosis process by providing information in Db2 accounting records for the longest wait time for a number of common suspension types. The following suspension types are included:

- IRLM lock suspensions
- Db2 internal latch suspensions
- Waits for Db2 synchronous or asynchronous I/O
- Waits for Db2 service tasks
- Waits for page latches

For more information, see [Chapter 11, “IFCID changes in Db2 13,” on page 111](#) and the DSNWMSGSG file.

IBM z16 group buffer pool (GBP) residency time

Starting in function level 100, two new statistics are added to relevant group buffer pool statistics storage areas:

- The weighted-average time a data area resides in a storage class before it is reclaimed, or 0 if the group buffer pool has not been reclaimed.
- The weighted-time a directory entry resides in a storage class before it is reclaimed, or 0 if the group buffer pool has not been reclaimed.

The new residency time statistics are supported only if Db2 13 runs in the following environment:

- z/OS 2.4 or later, with PTFs required for *cache residency time metrics* applied.
- IBM z16 or later processors.
- Coupling facility at CF Level 25 or higher. For more information, see [CFLEVEL and operating system level coexistence \(z/OS MVS Setting Up a Sysplex\)](#).

You can access these metrics with the IFCID record trace and the -DISPLAY GROUPBUFFERPOOL command. For more information, see "DSNB820I: Average residency times" in [DSNB750I \(Db2 Messages\)](#).

Subsystem parameter simplification

Function level 100 introduces changes to the default values for various subsystem parameters to match current best practices. It also removes a number of obsolete subsystem parameters. For a list of these changes, see [Chapter 12, “Subsystem parameter changes in Db2 13,” on page 117](#).

New-function APARs from Db2 12

New capabilities and enhancements from many new-function APARs in Db2 12 are built-in when you migrate to Db2 13. If you do not apply the PTFs in Db2 12, plan for the changes to take effect when you migrate to Db2 13. See the APARs with availability dates earlier than 2022-06 in [New-function APARs for Db2 12](#).

New-function APARs after Db2 13 GA

Many changes introduced in new-function APARs after the general availability of Db2 13 take effect immediately when you apply the PTF in function level 100 or higher. For more information, see [Chapter 2, “New-function APARs for Db2 13,” on page 33](#).

How to migrate to Db2 13 function level 100

To migrate to Db2 13 function level 100, complete the following tasks:

1. In Db2 12, identify and resolve incompatible changes and activate function level 510 (V12R1M510). You can run the pre-migration job DSNTIJPE in Db2 12 to identify the incompatible changes. For more information, see [Verify Db2 13 premigration activities and activate function level 510 in Db2 12 \(Db2 Installation and Migration\)](#).
2. Verify that every member was restarted with the fallback SPE applied in Db2 12.

Important: Inactive members that never started in Db2 12 with the fallback SPE ([APAR PH37108](#)) applied cannot start after the first data sharing member is migrated to Db2 13 at function level 100.

3. Migrate the Db2 subsystem or data sharing group to Db2 13, as described in [Migrating your Db2 subsystem to Db2 13 \(Db2 Installation and Migration\)](#) or [Migrating an existing data sharing group to Db2 13 \(Db2 Installation and Migration\)](#).

Related tasks

[Activating Db2 13 function level 500 or higher \(Db2 Installation and Migration\)](#)

[Adopting new capabilities in Db2 13 continuous delivery](#)

In Db2 13, function levels and application compatibility levels control the adoption of most new capabilities by Db2 subsystems and Db2 applications.

Related reference

[-ACTIVATE \(Db2\) \(Db2 Commands\)](#)

[New-function APARs for Db2 13](#)

New capabilities are introduced to Db2 13 through continuous delivery of APARs in a single service stream. Many APARs in Db2 13 deliver deactivated functional code to support future function levels. However, some APARs introduce their enhancements in Db2 13 regardless of the activated function level.

Part 2. Changes to plan for in Db2 13

Use this information when you are planning migration to Db2 13 and for planning to adopt new capabilities that Db2 13 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 13 or the activation of Db2 13 function levels. It also contains summaries of the changes that Db2 13 introduces, including separate sections for changes in the Db2 13 initial release and changes in higher function levels. It also summarizes function the Db2 13 no longer supports, and deprecated function that remains supported, but might be removed eventually.

Tip: New capabilities and enhancements from many new-function APARs in Db2 12 are built-in when you migrate to Db2 13. If you do not apply the PTFs in Db2 12, plan for the changes to take effect when you migrate to Db2 13. See the APARs with availability dates earlier than 2022-06 in [New-function APARs for Db2 12](#).

Related reference

[New, changed, and deleted codes \(Db2 Codes\)](#)

[New, changed, and deleted messages \(Db2 Messages\)](#)

Chapter 4. Incompatible changes in Db2 13

Before migrating your Db2 12 subsystem to Db2 13, applying maintenance in Db2 13, or activating higher function levels, familiarize yourself with incompatible changes that might impact your Db2 environment. Plan to resolve any such applicable incompatible changes that apply to your Db2 environment before or during the Db2 13 migration process, or before applying maintenance.

Function level	Explanation and possible impact	Actions to take
V13R1M504	Incremental binds might fail with different SQL codes When a statement-level invalidation occurs for a package that is bound with the DEPLEVEL(STATEMENT) option, the SYSPACKAGE.VALID column for the statement is marked with the value of 'S'. Statements in the package can be executed concurrently with autobind phase-in process, and any invalid statements go through incremental bind. If incremental binds fail due to prior invalidating changes to dependent objects, applications that previously failed with SQL code -904 might start receiving different SQL codes. For example, an incremental bind for a statement that depended on a dropped table might return -204 to the application.	Review the negative SQL codes in the application and take the appropriate actions as usual.
V13R1M502	New value of 'S' in the VALID column of SYSPACKAGE and SYSPACKCOPY catalog table An existing application might be marked with a value of 'S' in the VALID column of the SYSPACKAGE or SYSPACKCOPY table to indicate that the package has individual statements marked as invalid. There are no behavioral differences between how an invalidated package (VALID = 'N') and a partially invalidated package (VALID = 'S') are handled. The VALID value is reset to 'Y' in the same manner. However, if there are jobs or queries that look explicitly for 'N' in this column, these jobs might need to be modified to account for the new value of 'S'.	If you have jobs or queries that explicitly look for a value of 'N' in the VALID column of the SYSPACKAGE or SYSPACKCOPY catalog table, modify those jobs and queries to take into account the value of 'S', which indicates that statements within the package are invalid and the package needs to be rebound.
V13R1M100	Automatic rebind of plans and packages created before Db2 11 Migration-related <i>automatic binds</i> (also called "autobinds") occur in Db2 13 because it cannot use runtime	Rebind all packages that were last bound before Db2 11 in Db2 12, before you migrate to Db2 13. This action is required for any packages run in the last 18 months, before you can migrate

Function level	Explanation and possible impact	Actions to take
	<p>structures from a plan or package that was last bound in a release earlier than Db2 11. Plans and packages that were bound in Db2 12 can run in Db2 13, without the risk of migration-related autobinds. However, plans and packages that are bound in Db2 13 cannot run on Db2 12 members without an autobind in Db2 12.</p> <p>If you specify YES or COEXIST for the ABIND subsystem parameter, Db2 13 automatically rebinds plans and packages that were bound before Db2 11 when Db2 executes the packages.</p> <p>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 11. SQLCODE -908, SQLSTATE 23510 is returned for packages, and SQLCODE -923, SQLSTATE 57015 is returned for plans until they are rebound in Db2 13.</p>	<p>to Db2 13. For more information, see Verify Db2 13 premigration activities and activate function level 510 in Db2 12 (Db2 Installation and Migration).</p> <p>For more information about the impacts that migration-related automatic rebinds can have in your Db2 environment and actions that you can take to avoid them, see Rebind old plans and packages in Db2 12 to avoid disruptive autobinds in Db2 13 (Db2 Installation and Migration).</p>
V13R1M100	<p>Updated reset behavior for IFCID 402 counters</p> <p>Starting in Db2 12 or later with the PTF applied for APAR PH50729, Db2 resets IFCID 402 statistics counters immediately if the warning or exception threshold for the associated profile changes when you issue a START PROFILE command.</p> <p>In Db2 12, the new behavior is used only if the RESET_IFCID402 option is specified for the DDF_COMPATIBILITY subsystem parameter, which is removed in Db2 13.</p> <p>If the new behavior was used in Db2 12, or Db2 13 is installed with the PTF for APAR PH50729 applied, Db2 13 always uses the new behavior.</p> <p>However, in the following situations, the new behavior takes effect in Db2 13 only after you resubmit your customized copy of DSNTIJUZ and issue a -SET SYSPARM command, or stop and restart Db2:</p> <ul style="list-style-type: none"> The PTF for APAR PH50729 was applied in Db2 12 but the RESET_IFCID402 option was not 	<p>If you did not already start using the new behavior in Db2 12, take the following actions:</p> <ol style="list-style-type: none"> 1. Update any applications that use IFCID 402 counters to expect the new reset behavior. 2. Resubmit your customized copy of DSNTIJUZ. 3. Issue a SET SYSPARM command, or stop and restart Db2.

Function level	Explanation and possible impact	Actions to take
	<p>specified for DDF_COMPATIBILITY subsystem parameter.</p> <ul style="list-style-type: none"> Installation or migration to Db2 13 was completed before the PTF for APAR PH50729 is applied 	
V13R1M100	<p>Removed subsystem parameters</p> <p>Db2 13 enforces use of the default setting for several removed subsystem parameters. For the list of these subsystem parameters, see Chapter 12, “Subsystem parameter changes in Db2 13,” on page 117.</p> <p>A Db2 12 environment that uses a different setting than is enforced in Db2 13 might encounter problems at a migration, especially in Db2 data sharing coexistence, where some members run on Db2 12 while others are already migrated to Db2 13.</p>	<p>Before migrating to Db2 13:</p> <ol style="list-style-type: none"> Review the list of removed subsystem parameters, and the settings enforced by Db2 13. If your Db2 12 environment uses different settings, evaluate the potential impact of the new setting to your environment, and make any necessary changes. Ensure that your subsystem parameters module runs with the new settings before migrating to Db2 13.
V13R1M100	<p>Removal of DDF_COMPATIBILITY</p> <p>Db2 13 always operates as if this subsystem parameter setting is null.</p>	<p>If the DDF_COMPATIBILITY subsystem parameter is set to any non-null value in Db2 12, evaluate and make any necessary changes so that you can set it to null before migrating to Db2 13.</p>
V13R1M100	<p>Addition of MAX_UDF</p> <p>Starting in Db2 12 or later with the PTF applied for APAR PH44833, The MAX_UDF subsystem parameter controls the maximum number of user-defined external scalar functions that can run concurrently in a Db2 thread. The maximum value of MAX_UDF is 99999. Before the introduction of MAX_UDF, the maximum number of user-defined external scalar functions that could run concurrently in a Db2 thread was unlimited.</p>	<p>If an application contains SQL statements that invoke user-defined external scalar functions, and one of those SQL statements is rejected with SQLCODE -904 and reason code 00E70082, increase the MAX_UDF subsystem parameter value, or change the application to run fewer functions concurrently in a Db2 thread.</p>

Related concepts

[Changes to plan for in Db2 13](#)

Use this information when you are planning migration to Db2 13 and for planning to adopt new capabilities that Db2 13 introduces.

Related reference

[Deprecated function in Db2 13](#)

Certain features and capabilities that Db2 13 for z/OS supports are *deprecated*, meaning that their use is discouraged. Although they remain supported except as noted below in Db2 13, support is likely to be removed eventually.

Chapter 5. Storage changes in Db2 13

Db2 13 introduces changes to the storage configuration for a Db2 for z/OS installation.

More efficient cleanup for above-the-bar storage

Function level 100 introduces improvements to how Db2 manages and frees above-the-bar storage, especially to reduce the disruptive impact of issuing excessive IARV64 REQUEST(DISCARDDATA) service requests.

Db2 13 no longer issues the IARV64 REQUEST(DISCARDDATA) request during thread deallocation or at certain intervals of COMMIT, and enhanced storage management is no longer controlled by the REALSTORAGE_MANAGEMENT subsystem parameter, which is also removed. In Db2 13, the storage is returned to the memory object. A system-level timer drives contraction for the memory object to release unused frames back to z/OS. Also, Db2 13 periodically checks the available free frames before the LPAR starts to page (by using the z/OS calculations for available free frames and LO threshold). If this value becomes lower than 5 times the z/OS calculated OK threshold, the memory object contraction is triggered.

Improved storage monitoring and contraction

Function level 100 introduces the following enhancements to provide storage constraint relief:

- When below-the-bar Db2 storage consumption exceeds 64-percent threshold, Db2 automatically begins contraction of private storage pools.
- When extended common service area (ECSA) storage consumption exceeds the 85-percent threshold, Db2 automatically begins contraction of storage pools that are allocated in the ECSA.

In both cases, the storage contraction stops after storage consumption drops below the threshold.

This enhancement optimizes Db2 13 and improves its performance without changing how you configure, monitor, and use Db2.

Reduced ECSA storage use for distributed data facility (DDF) processing

Function level 100 reduces the amount of ECSA storage that is used for processing DDF threads to be equivalent to processing local threads. The previous recommendation was an extra 2 KB per DDF thread. For more information, see [Calculating the storage requirement for the extended common service area \(Db2 Installation and Migration\)](#).

Reduced ECSA storage for IFI buffers

Db2 13 reduces the use of ECSA storage for IFI buffers from a maximum of 50 MB to a fixed 8 MB.

Function level 100 reduces the use of ECSA storage for IFI buffers to a maximum of 25 MB. Then, after function level 500 is first activated, it is further reduced to 8 MB. The storage behavior that is introduced in function level 500 continues even if you later activate function level 100*.

To compensate for the reduction in ECSA storage, you must set aside an extra 50 MB for HVCOMMON and 25 MB for private storage. You can reduce the ECSA storage after function level 500 is activated and Db2 starts using the new storage pools. When Db2 uses the new storage pools, the use of ECSA for the retrieval of IFI records noticeably decreases. You can monitor use of the new storage pools by starting the statistics trace to collect IFCID 225. Then, you can check the SHARED / COMMON storage summary report in the formatted IDCID 225 SMF trace record.

For more information about ECSA storage requirements, see [Calculating the storage requirement for the extended common service area \(Db2 Installation and Migration\)](#).

Reduced agent local below-the-bar (BTB) storage

Starting in function level 100, Db2 supports a greater number of concurrent threads, by using above-the-bar (ATB) agent-local storage for statement text and attribute strings for dynamic SQL statements. In earlier releases, Db2 kept a copy of dynamic SQL statement text and attribute strings in agent local below-the-bar (BTB) storage while the statement is being prepared and executed.

For any specific thread, multiple dynamic SQL statements can be executing depending on the nesting level. The maximum length of an SQL statement is 2 MB, but much more storage can be allocated and the consumption of BTB storage could prevent the number of threads from scaling.

This enhancement optimizes Db2 13 and improves its performance without changing how you configure, monitor, and use Db2.

Dynamic management of CF lock storage by IRLM

With IRLM 2.3 at function level 50C or higher, which is included with Db2 13, IRLM can now invoke an existing capability in z/OS Sysplex Services for Data sharing (XES) to dynamically expand the coupling facility (CF) lock structure storage size. This new internal monitoring capability in IRLM can improve lock request processing throughput, by expanding the CF lock structure size to process lock requests, instead of rejecting them.

The existing XES monitoring of the CF lock structure use is defined in the coupling facility resource management (CFRM) policy as a threshold percentage value, and it is enabled with a default of 80% when it is not equal to zero. This monitoring retrieves statistics on the CF lock structure every 60 seconds when the storage usage is less than the threshold and every 30 seconds when the storage usage is equal to or greater than the full threshold. IRLM can determine the storage needed at a higher level of granularity than the existing monitoring by XES of CF structure, especially when a spike in locking activities results in rejection of lock requests due to insufficient Record List Entries (RLEs) even before z/OS has a chance to start the CF lock structure alteration.

The existing structure monitoring by XES handles storage contraction, and it contracts all eligible structures in the coupling facility by 10 percent in each cycle when the entire coupling facility is at or more than 90% full.

IRLM issues the following messages when it adjusts the CF lock structure storage size: [DXR189I](#) and [DXR190I](#).

Chapter 6. Command changes in Db2 13

You can use this information to plan for changes in Db2 and related commands at migration to Db2 13 and in continuous delivery.

Tip: The changes in the highest available function levels are listed first. If you are preparing for migration to Db2 13, start with [“Command changes in function level 100” on page 96](#).

Command changes in function levels 502 and higher

The following changes to commands take effect when you activate the specified function level in Db2 13. The changes are listed in function level order with the highest at the top.

Function level	Command	Change introduced	Incompatible change?
FL 502	BIND PACKAGE (DSN) (Db2 Commands)	New DEPLEVEL option, which determines if the current copy of a package records statement-level dependencies in addition to package-level dependencies.	No
FL 502	BIND SERVICE (DSN) (Db2 Commands)	New DEPLEVEL option, which determines if the current copy of a package records statement-level dependencies in addition to package-level dependencies.	No
FL 502	REBIND PACKAGE (DSN) (Db2 Commands)	New DEPLEVEL option, which determines if the current copy of a package records statement-level dependencies in addition to package-level dependencies.	No
FL 502	-MODIFY DDF (Db2) (Db2 Commands)	New option keywords: RQSTWLB, DFLTWLB for Db2 controlled sysplex workload balancing.	No
FL 504	-START TRACE (Db2) (Db2 Commands), -STOP TRACE (Db2) (Db2 Commands), -DISPLAY TRACE (Db2) (Db2 Commands)	The value of DEST(OPn) can be OP1 through OP16.	No

Command changes in function level 500

The following changes to commands take effect when you activate function level 500 in Db2 13. Activating function level 500 (V13R1M500) prevents coexistence with and fallback to Db2 12. Function level 500 is also the first opportunity for applications to use many of the new capabilities in Db2 13. However, new capabilities that depend on Db2 13 catalog changes remain unavailable.

Command	Change introduced	Incompatible change?
BIND PACKAGE(DSN) , BIND PLAN(DSN) , BIND SERVICE(DSN) , REBIND PACKAGE(DSN) , REBIND PLAN(DSN)	New options: OWNERTYPE(ROLE) or OWNERTYPE(USER) on the OWNER bind option .	No

Command	Change introduced	Incompatibl e change?
-SET LOG (Db2) (Db2 Commands)	New option: REMOVELOG.	No
-DISPLAY LOG (Db2) (Db2 Commands)	New option: DETAIL.	No

For more information about these changes, see [“Function level 500 \(for migrating to Db2 13 - May 2022\)”](#) on page 71.

Command changes in function level 100

The following changes to commands take effect when you migrate to function level 100 in Db2 13. Db2 starts at function level 100 (V13R1M100) during migration to Db2 13, and fallback and coexistence with Db2 12 in data sharing remain possible. Many new capabilities in Db2 13 remain unavailable.

Command	Change introduced	APAR	Incompatibl e change?
-DISPLAY ACCEL (Db2) (Db2 Commands)	If the DETAIL keyword is specified for multiple accelerators, Db2 13 issues a separate DSNX830I message is issued for each accelerator.	PH55472	No
-DISPLAY DYNQUERYCAPTURE (Db2) (Db2 Commands)	Users with the DISPLAY privilege have authority to issue this command.	PH56996 PH56997	No
-DISPLAY PROFILE (Db2) (Db2 Commands)	Users with the DISPLAY privilege have authority to issue this command.	PH56996 PH56997	No
FREE PACKAGE (DSN) (Db2 Commands)	This DSN subcommand can be used to free the inactive package copies of native REST services if an option other than ALL is specified for PLANMGMTSCOPE. Freeing of active packages for native REST services remains blocked with message DSNT264I.	PH54129	No
-MODIFY DDF (Db2) (Db2 Commands)	The ACCTNG option is supported to specify the granularity of accounting trace records for distributed applications. ACCTG(ALLCOMMIT) can be specified for additional granularity or ACCTG can be specified to return to the default behavior.	PH55241	No
-START TRACE (Db2) (Db2 Commands)	IFCID 0369 is added to statistics trace class 1. When -START TRACE (STAT) is issued without the CLASS parameter or with the CLASS(1) parameter, IFCID 0369 is activated by default. For compatibility with previous Db2 releases, statistics class 9, which contains IFCID 0369, is still available.	None	No

Command	Change introduced	APAR	Incompatibl e change?
-DISPLAY STATS (Db2) (Db2 Commands)	<p>The command output in message DSNT830I is updated such that TRAV. COUNT field returns a calculated traverse count value.</p> <p>A new FTB FACTOR field is added to return the calculated value that was previously returned in the TRAV. COUNT field, which can sometimes be negative if when index page-splits occur or index look-aside is used.</p> <p>If the output returns information about multiple indexes, descending order is used starting with the index with the highest FTB factor.</p>	PH51565	No

For more information about these changes, see [“Function level 100 \(for migrating to Db2 13 - May 2022\)”](#) on page 78.

Related information

[About Db2 and related commands \(Db2 Commands\)](#)

Chapter 7. SQL changes in Db2 13

You can use this information to plan for SQL changes at migration to Db2 13 and in continuous delivery.

Tip: The changes in the highest available function levels are listed first. If you are preparing for migration to Db2 13, start with [“SQL changes in function level 100”](#) on page 100.

SQL changes in function levels 502 and higher

The following SQL changes take effect when applications begin running at the specified application compatibility (APPLCOMPAT) level or higher. The changes are listed in APPLCOMPAT level order with the highest at the top.

APPLCOMPAT level	SQL element	Change introduced	Incompatible change?
FL 504	LISTAGG (Db2 SQL)	The ORDER BY clause can now be specified in a fullselect that contains an invocation of the LISTAGG built-in function.	No
FL 504	AI_COMMONALITY	New built-in function	No
FL 503	ALTER TABLE with ADD COLUMN for row change timestamp columns	When processing ALTER TABLE statements that specify ADD COLUMN for ROW CHANGE TIMESTAMP columns, Db2 now sets the corresponding value in the DEFAULTVALUE column value in the SYSIBM.SYSCOLUMNS catalog table to the timestamp of the ALTER TABLE statement.	No
FL 503	IN predicate (Db2 SQL)	Queries that specify IN list predicates with more than 32,767 (32K) elements can be run as accelerator-only queries in IBM Db2 Analytics Accelerator V7 for z/OS.	No
FL 503	SELECT INTO (Db2 SQL)	The <i>optimize-clause</i> , such as OPTIMIZE FOR 2 ROWS can be now specified in SELECT INTO statements.	No

SQL changes in function level 501

The following SQL changes take effect in Db2 13 for applications that run at application compatibility V13R1M501 or higher. Function level 501 (V13R1M501) is the first opportunity after migration to Db2 13 for applications to use new features and capabilities that depend on catalog changes in Db2 13.

SQL element	Change introduced	Incompatible change?
DEADLOCK_RESOLUTION_PRIORITY built-in global variable	New built-in global variable	No

For more information about these changes, see [“Function level 500 \(for migrating to Db2 13 - May 2022\)”](#) on page 71.

SQL changes in function level 500

The following SQL changes take effect in Db2 13 for applications that run at application compatibility V13R1M500 or higher. Activating function level 500 (V13R1M500) prevents coexistence with and fallback to Db2 12. Function level 500 is also the first opportunity for applications to use many of the new capabilities in Db2 13. However, new capabilities that depend on Db2 13 catalog changes remain unavailable.

SQL element	Change introduced	APAR	Incompatible change?
ALTER FUNCTION (compiled SQL scalar) (Db2 SQL)	New clauses: AS ROLE or AS USER on the PACKAGE OWNER clause	None	No
ALTER PROCEDURE (SQL - native) (Db2 SQL)	New clauses: AS ROLE or AS USER on the PACKAGE OWNER clause	None	No
CREATE FUNCTION (compiled SQL scalar) (Db2 SQL)	New clauses: AS ROLE or AS USER on the PACKAGE OWNER clause	None	No
CREATE PROCEDURE (SQL - native) (Db2 SQL)	New clauses: AS ROLE or AS USER on the PACKAGE OWNER clause	None	No
ALTER TABLE (Db2 SQL)	New clause: ALTER PARTITIONING TO PARTITION BY RANGE New concurrency behavior when the DATA CAPTURE clause is specified.	None	No
ALTER TABLE (Db2 SQL) with ALTER PARTITIONING TO PARTITION BY RANGE	Db2 no longer issues SQL code -20385 for certain pending data definition changes when you alter the partitioning scheme of a table from partition-by-growth (PBG) to partition-by-range (PBR).	PH51359	No
CREATE TABLESPACE (Db2 SQL)	The default value for the MAXPARTITIONS clause is changed from 256 to 254.	None	No
SET CURRENT LOCK TIMEOUT	New statement	None	No
CURRENT LOCK TIMEOUT special register	New special register	None	No
AI_ANALOGY	New built-in function	None	No
AI_SEMANTIC_CLUSTER	New built-in function	None	No
AI_SIMILARITY	New built-in function	None	No

For more information about these changes, see [“Function level 500 \(for migrating to Db2 13 - May 2022\)”](#) on page 71.

SQL changes in function level 100

The following SQL changes take effect in Db2 13 for applications that run at application compatibility V13R1M100 or higher. Db2 starts at function level 100 (V13R1M100) during migration to Db2 13, and fallback and coexistence with Db2 12 in data sharing remain possible. Many new capabilities in Db2 13 remain unavailable.

SQL element	Change introduced	APAR	Incompatibl e change?
Column names longer than 30 bytes	Function level 100 extends the maximum length of a column name from 30 bytes of EBCDIC, up to 128 bytes with limited support for using the longer column names. The longer column names can be used when the TABLE_COL_NAME_EXPANSION subsystem parameter setting is ON. Although you can now define a column with a name up to 128 bytes, column names with a length greater than 30 bytes of EBCDIC might be truncated on a character boundary. Column names returned in an SQLDA contain 30 bytes at most. APIs that do not use the SQLDA to obtain a column name might return complete column names.	None	No
CREATE TABLE statement	The PAGESET_PAGENUM subsystem parameter specifies the default for the PAGENUM option, and its default value is changed from ABSOLUTE to RELATIVE. see PAGE SET PAGE NUMBERING field (PAGESET_PAGENUM subsystem parameter) (Db2 Installation and Migration).	None	No
CREATE TABLESPACE statement	The PAGESET_PAGENUM subsystem parameter specifies the default for the PAGENUM option, and its default value is changed from ABSOLUTE to RELATIVE. see PAGE SET PAGE NUMBERING field (PAGESET_PAGENUM subsystem parameter) (Db2 Installation and Migration).	None	No
View privileges	DBAs with DBADM authority on databases can drop a view created for another user and select from a view without always requiring system level authority.	APARnnnnn	No

For more information about these changes, see [“Function level 100 \(for migrating to Db2 13 - May 2022\)” on page 78.](#)

Related reference

[Statements \(Db2 SQL\)](#)

[Reserved words in Db2 for z/OS \(Db2 SQL\)](#)

Chapter 8. Utility changes in Db2 13

You can use this information to plan for Db2 utilities changes at migration to Db2 13 and in continuous delivery.

Tip: The changes in the highest available function levels are listed first. If you are preparing for migration to Db2 13, start with [“Utility changes in function level 100”](#) on page 103.

Utility changes in function level 502 and higher

The following changes take effect when you activate function level 501 or higher in Db2 13. The changes are listed in function level order with the highest at the top.

Function level	Utility	Change introduced	Incompatible change?
FL 504	DSN1SDMP	In the START TRACE command, the DEST(OPn) keyword value can be OP1 through OP16.	No

Utility changes in function level 500

The following changes take effect when you activate function level 500 in Db2 13. Activating function level 500 (V13R1M500) prevents coexistence with and fallback to Db2 12. Function level 500 is also the first opportunity for applications to use many of the new capabilities in Db2 13. However, new capabilities that depend on Db2 13 catalog changes remain unavailable.

Utility	Change introduced	Incompatible change?
LOAD	New option: TABLESAMPLE SYSTEM	No
REORG INDEX	NOSYSUT1 is now the default when SHRLEVEL REFERENCE or CHANGE is specified.	No
REORG TABLESPACE	New option: TABLESAMPLE SYSTEM	No

For more information about these changes, see [“Function level 500 \(for migrating to Db2 13 - May 2022\)”](#) on page 71.

Utility changes in function level 100

The following changes take effect when you migrate to function level 100 in Db2 13. Db2 starts at function level 100 (V13R1M100) during migration to Db2 13, and fallback and coexistence with Db2 12 in data sharing remain possible. Many new capabilities in Db2 13 remain unavailable.

Utility	Change introduced	APAR	Incompatible change?
RECOVER	For certain object types, the RECOVER utility is enhanced to support space level recovery (where DSNUM ALL is specified or is the default), even when the image copies were created at the partition or piece level.	None	No
REPAIR	New option values: <ul style="list-style-type: none">• TYPE X'4002	None	No

Utility	Change introduced	APAR	Incompatibl e change?
	<ul style="list-style-type: none"> SUBTYPE 'X'000A' 		

For more information about these changes, see [“Function level 100 \(for migrating to Db2 13 - May 2022\)”](#) on page 78.

Related concepts

[Basic information about Db2 utilities \(Db2 Utilities\)](#)

Chapter 9. Catalog changes in Db2 13

You can use this information to plan for catalog changes at migration to Db2 13 and in continuous delivery.

Tip: The changes in the highest available function levels are listed first. If you are preparing for migration to Db2 13, start with [“Directory changes in function level 500” on page 107](#) and [“Catalog changes in catalog level V13R1M501” on page 106](#).

Db2 13 catalog levels

The *catalog level* of a data sharing group or subsystem indicates that a particular CATMAINT utility UPDATE LEVEL job was run on the Db2 catalog, and the data sharing group or Db2 subsystem is ready for the activation of certain function levels.

Db2 13 function levels use the following Db2 catalog levels:

- V13R1M504
- V13R1M501
- V13R1M100

For more information, see [Chapter 16, “Function levels and related levels in Db2 13,” on page 135](#).

Catalog changes for function levels V13R1M502 and higher

The following catalog changes take effect in Db2 13 when you tailor the Db2 catalog for activation of the specified function level. The changes are listed in function level order with the highest at the top.

Catalog level	Catalog object	Change introduced	Incompatible change?
V13R1M504	SYSIBM.SYSOBJEVENTS catalog table	FL 504 New table New table space: DSNCB06.SYSTSOEV New indexes: SYSIBM.DSNOEH01 and SYSIBM.DSNOEH02	No
V13R1M504	SYSPACKAGE table	The following column values are newly added or changed: <ul style="list-style-type: none">• OPERATIVE='R' is added indicates a package in rebind-advisory status because autobind phase-in has failed. If the package is allocated and any invalidated statements are executed, they are incrementally bound before execution. An explicit rebind is recommended to make the package valid and operative.• VALID='S' changes to indicate that the package can be allocated without waiting for the autobind process, and any invalidated statements are incrementally bound before execution.	No
V13R1M504	SYSPACKSTMTDEP table	FL 504 This table can contain additional records for phased-out package copies that are generated by autobind phase-in processing.	
V13R1M501	SYSPACKAGE catalog table (Db2 SQL)	FL 502 New possible column value for statement-level invalidation: VALID='S'	Yes

Catalog level	Catalog object	Change introduced	Incompatible change?
V13R1M501	SYSPACKCOPY catalog table (Db2 SQL)	FL 502 New possible column value for statement-level invalidation: VALID='S	Yes
V13R1M501	SYSCOLUMNS table	FL 503 The DEFAULTVALUE column value for newly added ROW CHANGE TIMESTAMP columns is set to a constant value derived from the timestamp of the ALTER TABLE statement that added the column.	No

Catalog changes in catalog level V13R1M501

The following catalog changes take effect in Db2 13 when you tailor the Db2 catalog for function level 501 activation.

Catalog object	Change introduced	Incompatible change?
DSNKDX02 catalog index	This index is removed.	No
SYSCOPY table	New column: EVENTID	No
SYSINDEXSPACESTATS table	Changed columns: <ul style="list-style-type: none"> • COPYCHANGES • EXTENTS • REORGAPPENDINSERT • REORGDELETES • REORGINSERTS • REORGPSEUDODELETES • SPACE • STATSDELETES • STATSINSERTS New columns: <ul style="list-style-type: none"> • REORGECSPLITS • REORGSPPLITIME • REORTOTALSPLITS 	No
SYSPACKAGE table	New column: DEPLEVEL Changed column: VALID	No
SYSPACKCOPY table	New column: DEPLEVEL Changed column: VALID	No
SYSPACKSTMT table	New column: VALID	No
SYSPACKSTMTCOPY table	New table New table space: DSNDDB06.SYSTSPSC New indexes:	No

Catalog object	Change introduced	Incompatibl e change?
	DSNKTX01 DSNKTX02	
SYSPACKSTMTDEP table	<p>New table New table space: DSNDDB06.SYSTSPSD New indexes:</p> <p>DSNKNX01 DSNKNX02</p> <p>Tip: FL 502 Db2 starts populating starts this table at application compatibility level V13R1M502 or higher.</p> <p>Tip: FL 504 This enhancement primarily introduces infrastructure to support a new capability that becomes available after you activate function level 504 or higher in Db2 13. For more information see Enabling autobind phase-in for packages invalidated at the statement level (Db2 Application programming and SQL).</p>	No
SYSTABLESPACESTATS catalog table (Db2 SQL)	<p>Changed columns:</p> <ul style="list-style-type: none"> • COPYCHANGES • EXTENTS • REORGDELETES • REORGINSERTS • REORGUPDATES • STATSDELETES • STATSINSERTS • STATSUPDATES 	No
SYSUTILITIES table	<p>New table New table space: DSNDDB06.SYSTSUTL New indexes:</p> <p>DSNULX01 DSNULX02</p>	No
SYSIBM.DSNSEQ_EVENTID sequence	A Db2-supplied sequence that is intended to be used to generate a value for the EVENTID column when a row is inserted into the SYSIBM.SYSUTILITIES catalog table.	No

For more information about these changes, see [“Function level 501 \(Db2 13 installation or migration - May 2022\)”](#) on page 67.

Directory changes in function level 500

Function level 500 introduces the following directory changes:

SPT01 and SYSLGRNX table spaces are converted to DSSIZE 256 GB

Starting in function level 500, the first time that the REORG TABLESPACE utility runs for the following directory objects, it converts the DSSIZE to 256 GB.

- DSNDB01.SPT01 to resolve issues that are related to the removal of the SPT01_INLINE_LENGTH subsystem parameter by APAR PH24358 in Db2 12.
- DSNDB01.SYSLGRNX in anticipation of future growth in this table for increasing workloads and conversions of non-UTS table space to UTS.

The conversion is automatic and does not require any special utility syntax. It updates the following Db2 catalog table values for each table space:

- The DSSIZE columns in SYSIBM.SYSTABLESPACE and SYSIBM.SYSTABLEPART are updated to 256G.
- A SYSCOPY record is inserted for the table space, with the following values to indicate that REORG changed the DSSIZE: ICTYPE = 'A', STYPE = 'D', TTYPE = '64G'.

If function level 100* is activated, already converted table spaces continue to use the larger DSSIZE, but the REORG utility does not convert unconverted table spaces.

Recovery to a point-in-time (PIT) before REORG converted the DSSIZE reverts the DSSIZE to 64GB. As always, if any one of the catalog or directory objects are recovered to a prior PIT, it is best to recover all catalog and directory objects to the same PIT.

Catalog changes in catalog level V13R1M100

No catalog changes take effect when you tailor the Db2 catalog for migration to Db2 13 function level 100.

Related reference

[Db2 catalog tables \(Db2 SQL\)](#)

Chapter 10. EXPLAIN table changes

Db2 13 introduces changes to the formats of certain EXPLAIN tables.

Tip: The changes in the highest available function levels are listed first. If you are preparing for migration to Db2 13, start with “Function level 100 EXPLAIN table changes” on page 109.

Important: Db2 13 introduces continuous delivery of new capabilities and enhancements in function levels. Most new capabilities become available only after activation of the Db2 13 function level that introduces them, or when applications run with the corresponding application compatibility level. For more information, see Chapter 20, “Activating Db2 13 function levels,” on page 147.

PSPI

For the current formats, and sample CREATE TABLE statements for PLAN_TABLE and the other EXPLAIN tables, see member DSNDESC 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 13 format during migration, or soon after migration. In Db2 13, the EXPLAIN function supports tables in Db2 13 or Db2 12 formats only. However, Db2 12 format EXPLAIN tables are deprecated. If you invoke EXPLAIN and Db2 12 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.

Function level 100 EXPLAIN table changes

The following changes to EXPLAIN tables take effect when you migrate to Db2 13 function level 100.

EXPLAIN table	Description of changes	APAR	Incompatibl e change?
PLAN_TABLE (Db2 Performance)	New column: AP_PLANID	PH48053	No
DSN_COLDIST_TABLE (Db2 Performance)	New column: AP_PLANID	PH48053	No
DSN_KEYTGTDIST_TABLE (Db2 Performance)	New column: AP_PLANID	PH48053	No
DSN_DETCOST_TABLE (Db2 Performance)	New column: AP_PLANID	PH48053	No
DSN_FUNCTION_TABLE (Db2 Performance)	New column: AP_PLANID	PH48053	No
DSN_PGRANGE_TABLE (Db2 Performance)	New column: AP_PLANID	PH48053	No
DSN_PGROUPTABLE (Db2 Performance)	New column: AP_PLANID	PH48053	No
DSN_PREDICAT_TABLE (Db2 Performance)	New column: AP_PLANID	PH48053	No
DSN_PTASK_TABLE (Db2 Performance)	New column: AP_PLANID	PH48053	No

EXPLAIN table	Description of changes	APAR	Incompatibl e change?
DSN_SORT_TABLE (Db2 Performance)	New column: AP_PLANID	PH48053	No
DSN_STAT_FEEDBACK (Db2 Performance)	New column: AP_PLANID	PH48053	No
DSN_STATEMENT_CACHE_T ABLE (Db2 Performance)	New columns: <ul style="list-style-type: none"> • STMT_HASHID2 • STMT_HASH2VER • AP_PLANID • AP_PLANHASH • AP_PLANHASHVER • CONNECTION_TYPE • CLIENT_USERID • CLIENT_APPLNAME • CLIENT_WRKSTNNAME 	PH48053	No
DSN_STATEMNT_TABLE (Db2 Performance)	New columns: <ul style="list-style-type: none"> • STMT_HASHID2 • STMT_HASH2VER • AP_PLANID • AP_PLANHASH • AP_PLANHASHVER • AP_SERVICE_DATA • CONNECTION_TYPE • CLIENT_USERID • CLIENT_APPLNAME • CLIENT_WRKSTNNAME 	PH48053	No
DSN_VIEWREF_TABLE (Db2 Performance)	New column: AP_PLANID	PH48053	No
DSN_FILTER_TABLE (Db2 Performance)	New column: AP_PLANID	PH48053	No
DSN_QUERYINFO_TABLE (Db2 Performance)	New column: AP_PLANID	PH48053	No

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 11. IFCID changes in Db2 13

Db2 13 introduces changes to various instrumentation facility component identifier (IFCID) trace fields.

Important: Db2 13 introduces continuous delivery of new capabilities and enhancements in function levels. Most new capabilities become available only after activation of the Db2 13 function level that introduces them, or when applications run with the corresponding application compatibility level. For more information, see [Chapter 20, “Activating Db2 13 function levels,”](#) on page 147.

Tip: IFCID numbers are often presented in Db2 and its documentation with three or four digits and leading zeros. For example, you might find: "IFCID 1," "IFCID 001," or "IFCID 0001." However, you can assume that these references each have the same meaning.

New trace records

PSPI

The following table gives an overview of new IFCIDs. Serviceability trace records are not included.

IFCID	Trace type and class	Mapping macro	Description
396	Statistics class 3, Performance class 6	DSNDQW05	Records information about abnormal index splits
437	Performance class 3	DSNDQW05	Records information about the SET CURRENT LOCK TIMEOUT SQL statement

PSPI

Changes to selected trace records

PSPI

The following table gives an overview of changes to specific IFCIDs. Changes to IFCID 106, the system parameters record, and changes to serviceability trace records are not included.

IFCID	Function level	APAR	Enhancement and description of changes
1	V13R1M100		<p>DBAT availability improvements:</p> <p>The value in field QDSTNDBA, which records the number of times that DBATs were created, might increase significantly after migration to Db2 13. In previous Db2 releases, field QDSTNDBA excluded DBATs that were created to replace disconnected, pooled DBATs that were terminated because they reached their reuse limit. Starting in Db2 13, the count in field QDSTNDBA includes all DBATs that were created.</p> <p>Fields are added to record:</p> <ul style="list-style-type: none">• The current number of DBATs (distributed server threads) that are active because the associated packages were bound with KEEP_DYNAMIC(YES).• The maximum number of DBATs that are active because the associated packages were bound with KEEP_DYNAMIC(YES).

IFCID	Function level	APAR	Enhancement and description of changes
			<ul style="list-style-type: none"> • The number of DBATs that were terminated since DDF was started. • The number of DBATs that were terminated because they remained in the pool longer than the value specified by the POOLINAC subsystem parameter. • The number of DBATs that were terminated because they were reused more times than the reuse limit. • The number of times that threads that were used by connections from the remote site were terminated because they remained in the pool longer than the value specified by the pool thread timeout value. • The number of times that threads that were used by connections from the remote site were terminated because the TCP/IP socket was closed due to a connection loss.
1	V13R1M100		<p>Increase number of latch classes to 64:</p> <p>The following changes are made:</p> <ul style="list-style-type: none"> • In the data that is mapped by DSNDQVLS, fields are added for latch classes 33 to 64. • Field QSST_P64DISYES is moved to the end of the data that is mapped by DSNDQSST.
1	V13R1M500	PH47626	<p>Improved DBAT status for MONITOR THREADS profiles in DISPLAY THREAD output:</p> <p>The following changes are made:</p> <ul style="list-style-type: none"> • The current number of DBATs that were suspended due to a profile exception. • The maximum number of DBATs that were suspended due to a profile exception.
2	V13R1M100	PH55153	<p>Changes to the maximum number of repeated sections in certain IFCID 2 records:</p> <p>The following changes are made:</p> <ul style="list-style-type: none"> • More than one IFCID 2 record can be generated for accelerator services statistics. The maximum number of QWS10RBO sections (accelerator services system statistics sections) in the initial IFCID 2 record is two. If additional Q8ST data exists, it is written in an additional IFCID 2 record that contains only data section 11 Q8ST data. • The maximum number of QWS10R7O sections (group buffer pool statistics sections) in the initial IFCID 2 record is decreased from 25 to 15.
2	V13R1M504	PH55497	<p>Minimize the impact of package invalidation</p> <p>Fields are added to record:</p> <ul style="list-style-type: none"> • The number of autobind phase-in attempts

IFCID	Function level	APAR	Enhancement and description of changes
			<ul style="list-style-type: none"> • The number of successful autobind phase-in processes • The number of autobind phase-in requests in the service task's queue
2, 3	V13R1M100		<p>Reduce RACF contention:</p> <p>Fields are added to record:</p> <ul style="list-style-type: none"> • The number of checks for the plan execute privilege that were made using the plan authorization cache and were unsuccessful because an applicable entry was not found in the cache. • The number of times that Db2 overwrote an authorization ID in the plan authorization cache.
2, 3	V13R1M500		<p>Application timeout and deadlock control:</p> <p>Fields are added to record:</p> <ul style="list-style-type: none"> • The number of times that a SET CURRENT LOCK TIMEOUT statement was executed. • The number of times that the CURRENT LOCK TIMEOUT special register was set from a profile table.
2, 3, 148	V13R1M504	PH55497	<p>Minimize the impact of package invalidation</p> <p>Fields are added to record:</p> <ul style="list-style-type: none"> • The number of incremental binds that occurred due to invalidated statements • The number of incremental binds with APREUSE that Db2 retried
3	V13R1M100		<p>Increase number of latch classes to 64:</p> <p>The following changes are made:</p> <ul style="list-style-type: none"> • Field QLLLLC is expanded from one byte to two bytes.
3	V13R1M100		<p>Accounting information on the longest wait times for common suspension types:</p> <p>A section is added to record the following thread-level wait time information:</p> <ul style="list-style-type: none"> • The longest wait for a lock, latch, or synchronous or asynchronous I/O (part 1 of the longest wait time information) • The longest service task wait time (part 2 of the longest wait time information) • The longest wait for a page latch (part 3 of the longest wait time information) <p>This information can simplify the task of diagnosing performance issues that are due to excessive wait times for resources.</p>
3	V13R1M100	PH46371	<p>Enhancement to accounting information on the longest wait times for common suspension types:</p>

IFCID	Function level	APAR	Enhancement and description of changes
			<ul style="list-style-type: none"> Field QLLLTYP is added to identify the various types of part 1 wait times.
3	V13R1M100	PH46372	<p>Enhancement to accounting information on the longest wait times for common suspension types:</p> <ul style="list-style-type: none"> Longest wait time information is written for accounting rollup records. Accounting rollup records are generated when subsystem parameter ACCUMACC is set to a value greater than 1.
3	V13R1M500	PH51892	<p>SQL Data Insights support:</p> <p>Fields are added to record:</p> <ul style="list-style-type: none"> Elapsed time, CPU time, and zIIP time that Db2 spent while processing SQL Data Insights functions. The number of times that SQL Data Insights functions were invoked.
21	V13R1M504	PH55497	<p>A flag is added for the locked resource type field to indicate that the lock is an autobind phase-in lock.</p>
51, 52, 56, 57, 148	V13R1M100		<p>Increase number of latch classes to 64:</p> <p>The following changes are made:</p> <ul style="list-style-type: none"> Fields QW0051LC, QW0052LC, QW0056LC, QW0057LC, and QW0148LC are expanded from one byte to two bytes.
172	V13R1M500		<p>Application timeout and deadlock control:</p> <p>A field is added to record:</p> <ul style="list-style-type: none"> For the process that is waiting for a resource, the source of the value that Db2 uses to determine the victim of a deadlock (worth value). <p>Existing long name fields are populated for client information such as workstation ID, user ID, and transaction ID.</p>
196	V13R1M500		<p>Application timeout and deadlock control:</p> <p>Fields are added to record the following information about a lock request that times out:</p> <ul style="list-style-type: none"> For the thread that requests a lock and times out, the source of the timeout interval. The source is the IRLMRWT subsystem parameter, the CURRENT LOCK TIMEOUT special register, or an IRLM internal value. For the thread that holds the lock, the source of the timeout interval that is set for the thread. The source is the IRLMRWT subsystem parameter, the CURRENT LOCK TIMEOUT special register, or an IRLM internal value. <p>Existing long name fields are populated for client information such as workstation ID, user ID, and transaction ID.</p>

IFCID	Function level	APAR	Enhancement and description of changes
230, 254	V13R1M100		<p>IBM z15 group buffer pool (GBP) residency time:</p> <p>Fields are added to record:</p> <ul style="list-style-type: none"> • The weighted average, in microseconds, of the elapsed time that a data area resides in a group buffer pool before the data area is reclaimed. • The weighted average, in microseconds, of the elapsed time that a directory entry resides in a group buffer pool before the directory entry is reclaimed.
319	V13R1M100	PH48764	A field (QW0319PA) is added to record the profile action ('W' for a warning or 'E' for an exception) that was taken if the IFCID 319 record was written because a KEYWORDS column value of MONITOR <i>product-type</i> CONNECTIONS FOR SECURITY was written in the DSN_PROFILE_ATTRIBUTES table.
319	V13R1M100	PH57811	A field is added to record the PROFILEID value when the QW0319PA field value is 'W' or 'E'.
376	V13R1M500		<p>Change of the default MAXPARTITIONS for partition-by-growth table spaces to 254:</p> <p>Function code 1315001 records when CREATE TABLESPACE or CREATE TABLE statements use a default MAXPARTITIONS value of 256 when the application compatibility level is V13R1M100, but use a default MAXPARTITIONS value of 254 when the application compatibility level is V13R1M500 or later. A record with function code 1315001 is written only when the application compatibility level is V13R1M100.</p>
389	V13R1M500	PH51565	A field is added to record the FTB factor, which is a calculated value that Db2 uses to identify indexes that might benefit from fast index traversal.
402	V13R1M100	PH50729	<p>The values of the following fields are reset to zero when a new profile is enabled, or an existing profile's warning or exception threshold is changed:</p> <ul style="list-style-type: none"> • QW0402TE • QW0402TQ • QW0402TF • QW0402TW • QW0402CE • QW0402CW • QW0402OE • QW0402OW

An additional location for the IFCID flat file (DSNWMSGGS)

You can now find DSNWMSGGS in either of the following locations:

- The most current version of DSNWMSGGS is available only for clients who have Db2 13 for z/OS licenses. The information is in a PDF file. To locate this information, see [Db2 13 for z/OS IFCID flat file \(DSNWMSGGS\)](#).

- An older version of DSNWMSGs is available in the *prefix.SDSNIVPD(DSNWMSGs)* data set. You can use the TSO or ISPF browse function to look at the field descriptions in *prefix.SDSNIVPD(DSNWMSGs)*, even when Db2 is down. If you prefer to look at the descriptions in printed form, you can use ISPF to print a listing of the data set.

◀ PSPI

Related concepts

[Types of Db2 traces \(Db2 Performance\)](#)

Related reference

[Trace field descriptions \(Db2 Performance\)](#)

Chapter 12. Subsystem parameter changes in Db2 13

You can use this information to plan for Db2 subsystem parameter changes at migration to Db2 13 and in continuous delivery.

Tip: The changes in the highest available function levels are listed first. If you are preparing for migration to Db2 13, start with [“Subsystem parameter changes in function level 100”](#) on page 118.

Subsystem parameter changes in function level 502 and higher

The following changes take effect when you activate function level 502 or higher in Db2 13.

Function level	Subsystem parameter	Change introduced	Incompatible change?
FL 504	UTILITY_HISTORY	New option: OBJECT	No
FL 504	MAX_CONCURRENT_PACKAGE_OPS	The value of this subsystem parameter now separately controls how many operations can run concurrently for two different types of automatic package bind operations: <ul style="list-style-type: none">Autobind requests (other than autobind phase-in requests), remote bind and rebind requests, CREATE TRIGGER statements, and CREATE PROCEDURE statements for packages.FL 504 Autobind phase-in requests. Consequently, the total number of autobind requests being processed concurrently can now be double the specified value.	No
FL 502	PACKAGE_DEPENDENCY_LEVEL	New subsystem parameter.	No
FL 503	QUERY_ACCELERATION_OPTIONS	New option field: 12 IN PREDICATE WITH MORE THAN 32K ELEMENTS	No

Subsystem parameter changes in function level 501

The following changes take effect when you activate function level 501 in Db2 13. Function level 501 (V13R1M501) is the first opportunity after migration to Db2 13 for applications to use new features and capabilities that depend on catalog changes in Db2 13.

Subsystem parameter	Change introduced	Incompatible change?
UTILITY_HISTORY	New subsystem parameter.	No

For more information about these changes, see [“Function level 501 \(Db2 13 installation or migration - May 2022\)”](#) on page 67.

Subsystem parameter changes in function level 500

The following changes take effect when you activate function level 500 in Db2 13. Activating function level 500 (V13R1M500) prevents coexistence with and fallback to Db2 12. Function level 500 is also the

first opportunity for applications to use many of the new capabilities in Db2 13. However, new capabilities that depend on Db2 13 catalog changes remain unavailable.

Subsystem parameter	Change introduced	Incompatibl e change?
<u>MXAIDTCACH</u>	New subsystem parameter.	No
<u>REORG_INDEX_NOSYSUT1</u>	This subsystem parameter no longer influences use of the NOSYSUT1 behavior for the REORG INDEX utility with SHRLEVEL REFERENCE or SHRLEVEL CHANGE. The default value is also changed to YES, and it is protected value.	No
<u>STATPGSAMP</u>	This subsystem parameter also applies to the collection of inline statistics when the LOAD or REORG TABLESPACE utilities run with the STATISTICS keyword.	No

For more information about these changes, see [“Function level 500 \(for migrating to Db2 13 - May 2022\)”](#) on page 71.

Subsystem parameter changes in function level 100

The following changes take effect when you migrate to function level 500 in Db2 13. Db2 starts at function level 100 (V13R1M100) during migration to Db2 13, and fallback and coexistence with Db2 12 in data sharing remain possible. Many new capabilities in Db2 13 remain unavailable.

Subsystem parameter	Change introduced	APAR	Incompatibl e change?
<u>AUTHEXIT_CACHEREFRESH</u>	If the value is set to ALL and the z/OS release is 2.5 or later, Db2 refreshes the entries in the plan authorization cache when a resource access on the plan object profile is changed in RACF and the access control authorization exit (DSNX@XAC) is active.	—	No
<u>DDF</u>	The default value is changed to AUTO.	—	No
<u>DSMAX</u>	The range is expanded with the maximum value increased from 200000 to 400000. The existing default of 200000 is retained.	—	No
<u>EDM_SKELETON_POOL</u>	The default value is changed from 51200 to 81920 in KB.	—	No
<u>EDMDBDC</u>	The default value is changed from 23400 to 40960 in KB.	—	No
<u>FTB_NON_UNIQUE_INDEX</u>	The default value is changed from NO to YES.	—	No
<u>IRLMRWT</u>	IRLMRWT is now online changeable to support more granular specification of the lock timeout value.	—	No
<u>LA_SINGLESEL_ISOCS_CDY</u>	New subsystem parameter.	PH49335	No
<u>MAXCONQN</u>	The default value is changed from OFF to ON.	—	No
<u>MAXCONQW</u>	The default value is changed from OFF to ON.	—	No

Subsystem parameter	Change introduced	APAR	Incompatible change?
MAXSORT_IN_MEMORY	The default value is changed from 1000 to 2000 (2 MB) in KB.	—	No
MAX_UDF	New subsystem parameter	PH44833	Yes
NUMLKTS	The default value is changed from 2000 to 5000.	—	No
NUMLKUS	The default value is changed from 10000 to 20000.	—	No
OUTBUFF	The default value is changed from 4000K to 104857600 (100MB).	—	No
PAGESET_PAGENUM	The default value is changed from ABSOLUTE to RELATIVE.	—	No
SPREG_LOCK_TIMEOUT_MAX	New subsystem parameter	—	No
SRTPOOL	The default value is changed from 10000 to 20000 (20MB) in KB.	—	No
STATIME_DDF	New subsystem parameter	PH56228	No
STATIME_MAIN	The default value is changed from 60 to 10. APAR PH56228 removes the control of the interval of location statistics trace records, IFCIDs 365, 411, and 412 from STATIME_MAIN when STATIME_DDF does not equal zero.	—	No
TABLE_COL_NAME_EXPANSION	New subsystem parameter.	—	No
ACCELERATION_OPTIONS field (QUERY_ACCEL_OPTIONS subsystem parameter) (Db2 Installation and Migration)	New fields on panel DSNTIP8A: <ul style="list-style-type: none"> • 12 IN PREDICATE WITH MORE THAN 32K ELEMENTS (TBD) • 13 ENABLE ACCELERATOR SPECIFIC RESULTS (PH48480) 	PH48480, TBD	No

For more information about these changes, see [“Function level 100 \(for migrating to Db2 13 - May 2022\)” on page 78.](#)

Removed subsystem parameters in Db2 13 (function level 100 and higher)

The following table 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.

Important: For best results, check the setting used in your Db2 12 environment, especially in data sharing environments. If the current setting does not match the setting listed in the following table, evaluate whether any other changes are needed to accept the new-behavior settings, before migrating to Db2 13.

Subsystem parameter	Setting used in Db2 13	Description in earlier releases	Incompatible change?
AUTHCACH	4K	Specifies the size (in bytes per plan) of the authorization cache that is to be used if no CACHESIZE is specified on the BIND PLAN subcommand.	No
DDF_COMPATIBILITY	NULL	Controls certain characteristics of a connection between a client application and a Db2 for z/OS data server.	Yes
DSVCI	YES	Controls whether Db2-managed data sets that are created by CREATE TABLESPACE or CREATE INDEX statements are to have variable VSAM control intervals.	No
EXTRAREQ	100	Limits the number of extra DRDA query blocks that Db2 can request from a remote DRDA server.	No
EXTRSRV	100	Limits the number of extra DRDA query blocks that Db2 can return to a DRDA client	No
HONOR_KEEPPDICTIONARY	NO	Specifies whether Db2 honors the LOAD and REORG parameter KEEPPDICTIONARY when tables are converted between basic row format and reordered row format.	No
IMMEDWRI	NO	Determines when updates to group buffer pool-dependent buffers are to be written to the coupling facility.	No
IX_TB_PART_CONV_EXCLUDE	YES	Specifies whether to exclude trailing columns from the table-controlled partitioning keys when table spaces are converted from index-controlled partitioning to table-controlled partitioning.	No
MAXARCH	10000	Controls the maximum number of archive log volumes that are to be recorded in the BSDS.	No
MAXTYPE1	0	Determines the number of inactive DBATs that Db2 is to allow.	No
OPT1ROWBLOCKSORT	DISABLE	Specifies whether Db2 explicitly blocks sort operations when the OPTIMIZE FOR 1 ROW clause is specified on a query.	No
PARA_EFF	50	Controls the efficiency that Db2 assumes for parallelism when Db2 chooses an access path.	No
PLANMGMTSCOPE	STATIC	Specifies the types of SQL statements for applying the PLANMGMT subsystem parameter setting.	No

Subsystem parameter	Setting used in Db2 13	Description in earlier releases	Incompatible change?
REALSTORAGE_MANAGEMENT	AUTO “1” on page 121	Specifies whether Db2 should manage real storage consumption. Important: Keep your current setting in Db2 12. See “1” on page 121.)	No
RESYNC	2	Specifies the time interval, in minutes, between resynchronization periods.	No
SUBQ_MIDX	ENABLE	Specifies whether to enable or disable multiple index access on some non-Boolean uncorrelated subquery predicates.	No
TRACSTR	NO	Specifies whether the global trace starts automatically when Db2 starts.	No

Notes:

1. If you currently use a different setting, do not change the REALSTORAGE_MANAGEMENT setting to AUTO in Db2 12.

Db2 13 uses enhanced automatic behavior for real storage management, which differs from the thread-based discard processing used with REALSTORAGE_MANAGEMENT=AUTO in Db2 12 or earlier.

Db2 13 manages real storage discard processing at the system level to avoid z/OS RSM serialization from discard requests.

Related reference

[Directory of subsystem parameters, panel fields, and application default values \(Db2 Installation and Migration\)](#)

Chapter 13. The extended 10-byte RBA and LRSN in Db2 13

As with earlier releases, Db2 13 continues to use the extended 10-byte RBA and LRSN format that was first introduced in Db2 11. No new changes that are related to the extended 10-byte RBA and LRSN format are introduced by Db2 13, and the BSDS conversion was already required at migration to Db2 12. However, if you still have page sets yet to be converted to use extended format, you might sometimes encounter problems with them. This information is included as a reminder to develop plans to convert any page sets that still use the basic 6-byte format.

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 beginning of the value, as shown in the following table.

6-byte RBA value:	10-byte RBA value:
112233445566	00000000112233445566

Conversion LRSN values

A 6-byte LRSN value is converted to a 10-byte value by adding one zero byte to the beginning and 3 bytes added to the end of the value, as shown in the following table.

6-byte LRSN value:	10-byte LRSN value:
112233445566	00112233445566000000

The 3 bytes on the right side 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

Extended RBA and LRSN values are externalized in the following contexts before objects are converted:

Messages

In Db2 13, all messages use 10-byte RBA and LRSN values, 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, 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.

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.

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 [SYSLOGRX 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. The BSDS conversion was required at migration to Db2 12, so the BSDS always uses the extended 10-byte format in Db2 12 or later.



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, as soon as possible.



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.

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 more information about the RBA and LRSN logging limits, see [What to do before RBA or LRSN limits are reached \(Db2 Administration Guide\)](#).

Catalog table columns

The Db2 catalog and directory contain RBA and LRSN information in several tables.

Catalog and directory columns that contain RBA or LRSN values use 10-byte format. 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.

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.

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.

Utilities

In Db2 13, 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.

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 or later (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 when you are ready.



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, as soon as possible.

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.

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 more information about the RBA and LRSN logging limits, see [What to do before RBA or LRSN limits are reached \(Db2 Administration Guide\)](#).

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 00100000002FFFFFFF 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 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 14. Function that Db2 13 no longer supports

Certain features that were supported in Db2 12 are no longer supported in Db2 13.

Function level or APAR	Removed support	Behavior in Db2 13 and alternative
V13R1M100	Various subsystem parameters are removed in Db2 13.	See Adjust subsystem parameter settings for parameters removed in Db2 13 (Db2 Installation and Migration) .

Deprecated function in Db2 13

Certain capabilities that Db2 13 supports are deprecated, meaning that their use is discouraged. Although they currently remain supported in Db2 13, 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.

For more information, see [Chapter 15, “Deprecated function in Db2 13,” on page 129](#).

Chapter 15. Deprecated function in Db2 13

Certain features and capabilities that Db2 13 for z/OS supports are *deprecated*, meaning that their use is discouraged. Although they remain supported except as noted below in Db2 13, support is likely to be removed eventually.

Avoid creating new dependencies that rely on deprecated features and capabilities, and develop plans to remove any dependencies on them.

Table 1. *Deprecated features and capabilities in Db2 13*

Deprecated function	Recommended alternative	Support removed
Basic row format table spaces	<p>Use reordered row format.</p> <p>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:</p> <ul style="list-style-type: none">• 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.	—
CMTSTAT subsystem parameter	Set the CMTSTAT value to INACTIVE. Although, the ACTIVE setting can sometimes provide limited CPU performance advantages, they are generally outweighed by increased use of other system resources and the lack of scalability for distributed application workloads. For the recommended alternatives to the ACTIVE setting, see Enabling KEEP DYNAMIC refresh for DBATs (Db2 Performance) and Enabling high-performance DBATs (Db2 Performance) .	—

Table 1. Deprecated features and capabilities in Db2 13 (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.	—
CHECK_FASTREPLICATION subsystem parameter	Use the default value REQUIRED.	—
DEPLOY option of the BIND PACKAGE command	Deploy native SQL procedures and compiled SQL scalar functions to multiple environments by issuing the same CREATE or ALTER statements separately in each Db2 environment.	—
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.	—
DSNRLMTxx 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) .	—
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.	—
SYSPROC.DSNWZP	Use the SYSPROC.ADMIN_INFO_SYSPARM stored procedure.	—
External SQL procedures	Use native SQL procedures, which are more fully supported, easier to maintain, and typically perform better than external SQL procedures. For more information, see Migrating an external SQL procedure to a native SQL procedure (Db2 Application programming and SQL)	—

Table 1. Deprecated features and capabilities in Db2 13 (continued)

Deprecated function	Recommended alternative	Support removed
Hash-organized tables	Alter tables to drop hash organization, and , create indexes to support fast index traversal in Db2 12 or higher. For more information, see Fast index traversal (Db2 Performance) .	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.
LOAD utility IDENTITY_OVERRIDE option	Use the OVERRIDE(IDENTITY) option.	—
LOAD utility PERIOD_OVERRIDE option	Use the OVERRIDE(SYSTEMPERIOD) option.	—
LOAD utility TRANSID_OVERRIDE option	Use the OVERRIDE(TRANSID) option.	—
NEWFUN SQL processing option	Use SQLLEVEL. NEWFUN is ignored if SQLLEVEL is specified.	—
ODBC 2.0 function	See Deprecated ODBC functions (Db2 Programming for ODBC) .	—
PassTickets for RACF-protected user IDs.	Use client certificate authentication.	—
PREVENT_NEW_IXCTRL_PART subsystem parameter	Use the default value YES.	—
REORG INDEX utility LEAFDISTLIMIT 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 UNLOAD ONLY option	Use the UNLOAD utility.	—
REORG TABLESPACE utility UNLOAD PAUSE option	Use the UNLOAD utility FORMAT INTERNAL option.	—
REPAIR VERSIONS utility	Use the REPAIR CATALOG utility.	—

Table 1. Deprecated features and capabilities in Db2 13 (continued)

Deprecated function	Recommended alternative	Support removed
SNA communication methods, including the VTAM interface	Use TCP/IP communication only. You can disable SNA communication by specifying a value in the DB2 TCP/IP IPNAME field on panel DSNTIPR.	—
DB2XML.SOAPHTTPPC supplied user-defined function	Use the DB2XML.SOAPHTTPNC user-defined function.	—
DB2XML.SOAPHTTPPV supplied user-defined function	Use the DB2XML.SOAPHTTPNV user-defined function.	—
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.	—
Non-UTS table spaces for base tables, including segmented (non-UTS), partitioned (non-UTS), and simple table spaces.	<p>Use partition-by-growth or partition-by-range universal table spaces instead.</p> <p>In Db2 12, Packages bound with APPLCOMPAT(V12R1M504) or higher cannot create objects of the following types:</p> <ul style="list-style-type: none"> • 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 <p>You cannot create new simple table spaces in any supported Db2 release.</p>	—

Related tasks



[Preparing your system to install or migrate to Db2 13 \(Db2 Installation and Migration\)](#)

Part 3. Adopting new capabilities in Db2 13 continuous delivery

In Db2 13, function levels and application compatibility levels control the adoption of most new capabilities by Db2 subsystems and Db2 applications.

About this task

Function levels are specified by strings that correspond to the Db2 version, release, and maintenance value. The format is *VvvvRrMmmm*, where *vv* is the version, *r* is the release, and *mmm* is the modification level. For example, V13R1M504 identifies function level 504. For a list of all available function levels in Db2 13, see [Chapter 3, “Db2 13 function levels,”](#) on page 55. Often function level identifiers are abbreviated. For example, “function level 504” refers to V13R1M504.

Tip:  You can determine the catalog level and function level for a Db2 subsystem or data sharing group, and the code levels of individual subsystems or members, by issuing DISPLAY GROUP commands. For more information, see [Chapter 17, “Determining the Db2 code level, catalog level, and function level,”](#) on page 139 .

Procedure

To manage the adoption of new capabilities in Db2 13, use the following overall process:

1. Apply maintenance to bring the Db2 subsystem to the required code level or higher.

Tip: Apply the maintenance for a code level well before you tailor the catalog level or activate a function level. By doing so, you can verify that Db2 can continue run at the required code level, while you still have the opportunity to identify and remove any problematic maintenance items.

Important: Do not attempt to start Db2 at any code level that is lower than the highest ever activated function level, even at the lower star (*) function level. Activate a function level only after you are satisfied that Db2 can continue to run at the required code level.

2. If necessary, update the Db2 catalog.

You can use a single CATMAINT job that specifies the target function level. If the target function level requires multiple catalog level updates, the CATMAINT job processes each update in sequential order. If a later update in the sequence fails, the previous successful updates do not roll back, and the catalog level remains at the highest level reached. If that occurs, you can correct the reason for the failure and resubmit the same CATMAINT job. Some function levels do not require catalog changes.

Db2 13 function levels use the following Db2 catalog levels:

- V13R1M504
- V13R1M501
- V13R1M100

Important: Do not attempt to start Db2 at a lower code level after any part of the CATMAINT job for a higher function level completes. Run the CATMAINT job only after you are satisfied that Db2 can continue to run at the necessary code level. The code to tolerate catalog changes is contained in the code level that delivers the CATMAINT job.

3. Activate the higher function level.

Some new capabilities and enhancements become available immediately. Optimization enhancements become available after the next full prepare of the SQL statements. The application compatibility level of each application continues to control the use of new SQL capabilities.

4. When you are ready for applications, and objects such as routines or triggers, to use new SQL capabilities of the higher function level, rebind or alter them at the higher application compatibility level. Do this only after you are satisfied that Db2 13 is stable at the higher function level.

You might need to adjust your applications for incompatible changes before they can run at the higher application compatibility level.

Tip: Do not raise the default application compatibility level of the Db2 subsystem immediately after migrating or activating a new function level. Instead, wait until applications have been verified to work correctly at the higher function level, and any incompatibilities have been resolved. For details, see [Enabling default application compatibility with function level 500 or higher \(Db2 Application programming and SQL\)](#).

Related concepts

[What's new in Db2 13](#)

Db2 13 for z/OS brings leading-edge innovation to reinforce Db2 for z/OS as a foundation for enterprise computing within the hybrid cloud world.

[Application compatibility levels in Db2 \(Db2 Application programming and SQL\)](#)

Related tasks

[Managing application incompatibilities \(Db2 Application programming and SQL\)](#)

Related reference

[Db2 13 function levels](#)

New Db2 capabilities and enhancements are continuously delivered in a single maintenance stream as the code becomes ready. You can activate the new capabilities in a data sharing group or Db2 subsystem after a function level is delivered. A *function level* corresponds to a single PTF that enables the activation of a specific set of enhancements that shipped in previous prerequisite or co-requisite PTFs. The activation of a function level results in the activation of all lower function levels.

Related information



[PI70406: Add ACTIVATE mode to the Db2 installation CLIST](#)

Chapter 16. Function levels and related levels in Db2 13

Enhancements to Db2 13 are enabled for use when you activate function levels. Before you can activate a function level, the Db2 subsystem, or all Db2 data sharing members, must be at a sufficient code level, and the Db2 catalog must be updated to the required Db2 catalog level.

Each *function level* corresponds to a single APAR that enables a set of enhancements that were previously delivered in the service stream. A particular function level might enable one or several enhancements.

In most cases, you can activate a higher function level without separately activating each lower function level above the currently activated function level. However, activating a higher function level also activates all capabilities that are introduced by all function levels lower than the one being activated.

Tip:  You can determine the catalog level and function level for a Db2 subsystem or data sharing group, and the code levels of individual subsystems or members, by issuing DISPLAY GROUP commands. For more information, see [Chapter 17, “Determining the Db2 code level, catalog level, and function level,” on page 139](#) .

Function level identifiers

In most cases, code levels, catalog levels, function levels, and application compatibility levels are specified in commands and message output by nine-character strings that correspond to the Db2 version, release, and modification value. However, descriptions of function levels in documentation often refer only to the modification part of the values.

The format is *VvvRrMmmm*, where *vv* is the version, *r* is the release, and *mmm* is the modification level. For example, V13R1M504 identifies function level 504. For a list of all available function levels in Db2 13, see [Chapter 3, “Db2 13 function levels,” on page 55](#).

Code levels

The *code level* of a Db2 subsystem or data sharing member indicates that the necessary APAR and any prerequisite new function code, defect fixes, and other service items for a corresponding function level are applied. Because new function levels are delivered in the same service stream as other maintenance items, the code level is likely to increase as you routinely apply maintenance to a subsystem or member. If you proactively apply maintenance, you can expect the code level to be higher than the catalog level or function level as you prepare to adopt of new Db2 capabilities.

If you remove maintenance items that support or are otherwise related to a code level, Db2 reverts to a lower code level. However, you cannot start Db2 at a lower code level after tailoring the catalog to a higher catalog level or activating a higher function level. For this reason, it is essential that you tailor the catalog at a higher catalog level or activate a function level only after you are certain that Db2 can continue to run at the corresponding code level.

Tip: Apply the maintenance for a code level well before you tailor the catalog level or activate a function level. By doing so, you can verify that Db2 can continue run at the required code level, while you still have the opportunity to identify and remove any problematic maintenance items.

In a data sharing group, each member can be at a different code level. However a function level is always activated at the group level. That is, if any data sharing member is not at the minimum required code level when you attempt to activate a function level, the ACTIVATE command fails. The DSNU757I message output indicates the current and required code levels.

Each code level is identified by the same identifier as the function level that it enables. The format is *VvvRrMmmm*, where *vv* is the version, *r* is the release, and *mmm* is the modification level.

Important: Do not attempt to start Db2 at any code level that is lower than the highest ever activated function level, even at the lower star (*) function level. Activate a function level only after you are satisfied that Db2 can continue to run at the required code level.

In DISPLAY GROUP command output, the DB2 LVL column indicates the code level of each data sharing member or subsystem in a six-character string that contains the Db2 version, release, and modification values. The format is *vvrmmm*, where *vv* is the version, *r* is the release, and *mm* is the modification level.

Db2 13 catalog levels

The *catalog level* of a data sharing group or subsystem indicates that a particular CATMAINT utility UPDATE LEVEL job was run on the Db2 catalog, and the data sharing group or Db2 subsystem is ready for the activation of certain function levels.

Each function level requires a specific catalog level. However, not every function level requires a new catalog level. If the catalog is not at the minimum required level when you attempt to activate a function level, the ACTIVATE command fails. The message output indicates the current and required catalog levels. You can also use the DISPLAY GROUP command to identify the current Db2 catalog level.

Db2 13 function levels use the following Db2 catalog levels:

- V13R1M504
- V13R1M501
- V13R1M100

Tip: When you update the Db2 catalog level, you can specify either the target function level or the catalog level. If you specify the function level, Db2 determines the appropriate target catalog level, and the CATMAINT utility completes any required catalog level updates in sequential order. For more information, see [CATMAINT \(Db2 Utilities\)](#).

The catalog level is updated when you submit a CATMAINT utility job by tailoring and running the DSNTIJTC sample job. You can use a single CATMAINT job that specifies the target function level. If the target function level requires multiple catalog level updates, the CATMAINT job processes each update in sequential order. If a later update in the sequence fails, the previous successful updates do not roll back, and the catalog level remains at the highest level reached. If that occurs, you can correct the reason for the failure and resubmit the same CATMAINT job.

Whereas different data sharing members can be at different code levels in a data sharing group, a data sharing group has a single catalog level.

Important: Do not attempt to start Db2 at a lower code level after any part of the CATMAINT job for a higher function level completes. Run the CATMAINT job only after you are satisfied that Db2 can continue to run at the necessary code level. The code to tolerate catalog changes is contained in the code level that delivers the CATMAINT job.

Each catalog level is identified by the same identifier as the lowest function level that requires it. The format is *VvvRrMmmm*, where *vv* is the version, *r* is the release, and *mmm* is the modification level. For example function level 504 requires catalog level V13R1M504.

When you first migrate to Db2 13 the catalog level is V13R1M100. The structure of the catalog does not change until you tailor the catalog for function level 501, as described in [Activating Db2 13 function level 500 or higher \(Db2 Installation and Migration\)](#).

To find when the catalog level changed, you can check the SYSIBM.SYSLEVELUPDATES catalog table or check for message DSNG014I on the console.

Function levels

A *function level* enables a particular set of new Db2 capabilities and enhancements that were previously delivered in the single continuous stream of Db2 code. It includes code that supports new capabilities, defect fixes, and preventive service items. Before you can use the new capabilities of a function level, you

must activate the function level, or a higher function level. Activation of a function level implies activation of the capabilities that are introduced by all lower function levels.

Important: Do not attempt to start Db2 at any code level that is lower than the highest ever activated function level, even at the lower star (*) function level. Activate a function level only after you are satisfied that Db2 can continue to run at the required code level.

In data sharing groups, function levels are activated at the group level. That is, if any data sharing member is not at the minimum required code level when you attempt to activate a function level, the ACTIVATE command fails. The DSNU757I message output indicates the current and required code levels.

To find when the function level changed, you can check the SYSIBM.SYSLEVELUPDATES catalog table or check for message DSNG014I on the console.

Before applications can use new SQL capabilities that a function level introduces, the applications must run at the equivalent application compatibility level or higher.

Application compatibility levels

You can use the *application compatibility* level of applications, and objects such as routines or triggers, to control the adoption and use of new and changed SQL capabilities that are introduced in function levels. Generally, applications, and routines or triggers, cannot use new or changed SQL capabilities unless the effective application compatibility level is equivalent to or higher than the function level that introduced the changes. The application compatibility level applies to most SQL statements, including data definition statements (such as CREATE and ALTER statements) and data control statements (such as GRANT and REVOKE statements).

The corresponding function level or higher must be activated when you bind packages at an application compatibility level. However, if you activate a lower function level (or * function level), applications can continue to run with the higher application compatibility level. To prevent the continued use of SQL capabilities introduced in the higher function level, you must also modify the application and change the effective application compatibility level to the lower level.

Tip: Extra program preparation steps might be required to increase the application compatibility level for applications that use data server clients or drivers to access Db2 for z/OS. For more information, see [Setting application compatibility levels for data server clients and drivers \(Db2 Application programming and SQL\)](#).

Tip: Do not raise the default application compatibility level of the Db2 subsystem immediately after migrating or activating a new function level. Instead, wait until applications have been verified to work correctly at the higher function level, and any incompatibilities have been resolved. For details, see [Enabling default application compatibility with function level 500 or higher \(Db2 Application programming and SQL\)](#).

Application compatibility levels for Db2 13 are identified by the same identifier as the corresponding function level. The format is *VvvRrMmmm*, where *vv* is the version, *r* is the release, and *mmm* is the modification level.

Db2 13 supports the following application compatibility levels in most contexts:

VvvRrMmmm

Compatibility with the behavior of the identified Db2 function level. For example, V13R1M504 specifies compatibility with the highest available Db2 13 function level. The equivalent function level or higher must be activated. For a list of supported Db2 13 function levels, see [Chapter 3, “Db2 13 function levels,” on page 55](#). Db2 13 also supports values from Db2 12. See [Db2 12 function levels](#).

Tip: Extra program preparation steps might be required to increase the application compatibility level for applications that use data server clients or drivers to access Db2 for z/OS. For more information, see [Setting application compatibility levels for data server clients and drivers \(Db2 Application programming and SQL\)](#).

V12R1

Compatibility with the behavior of Db2 12 function level 500. This value has the same result as specifying V12R1M500.

V11R1

Compatibility with the behavior of Db2 11 new-function mode.

V10R1

Compatibility with the behavior of DB2 10 new-function mode.

For more information about application compatibility levels, see [Application compatibility levels in Db2 \(Db2 Application programming and SQL\)](#).

Star (*) function levels

You might sometimes activate a lower function level, called a *star (*) function level* after you activate a higher function level. Any function level lower than the highest function level that was ever activated is always a star (*) function level.

The output from DISPLAY GROUP and ACTIVATE commands identify star (*) function levels by the function level identifier followed by an asterisk, hence the name. For example, assume that you activate function level 500 after function level 501 was previously activated. V13R1M500* in the message output indicates that the Db2 data sharing group or subsystem is at function level 500* (you might say, "function level 500 star").

Important: Activating a lower star (*) function level does not enable you to remove maintenance and start Db2 at any code level lower than the catalog level or highest ever activated function level.

Activating a lower star (*) function level by itself does not immediately disable all use of capabilities at higher function levels. Instead, it provides flexibility for limiting or disabling the use of capabilities at higher function levels, while the problems encountered in higher function levels are resolved.

At a star (*) function level, Db2 continues to tolerate objects, packages, and structures that were created or bound at higher function levels. Also, in any context where the effective application compatibility level remains at the higher level, new SQL capabilities from the higher level can still be used. For packages, they can still be executed, rebound, and automatically bound. However, an explicit bind of such packages succeeds only when the APPLCOMPAT bind option is equivalent to the activated star (*) function level or lower. Similar rules apply for the application compatibility levels of native SQL procedures, compiled SQL scalar functions, and advanced triggers. The result is that applications that use capabilities at a higher function level can continue to do so if they are not related to the reason for activating the lower function level. To stop the use of all SQL capabilities at the higher function level, you must also set all effective application compatibility levels at the lower level.

Related concepts

[Application compatibility levels in Db2 \(Db2 Application programming and SQL\)](#)

Related reference

Db2 13 function levels

New Db2 capabilities and enhancements are continuously delivered in a single maintenance stream as the code becomes ready. You can activate the new capabilities in a data sharing group or Db2 subsystem after a function level is delivered. A *function level* corresponds to a single PTF that enables the activation of a specific set of enhancements that shipped in previous prerequisite or co-requisite PTFs. The activation of a function level results in the activation of all lower function levels.

[-DISPLAY GROUP \(Db2\) \(Db2 Commands\)](#)

Related information

[DSN7100I \(Db2 Messages\)](#)

[DSNG014I \(Db2 Messages\)](#)

[Video: Db2 for z/OS—Delivering New Capabilities Faster \(YouTube: 1:33:25\)](#)

Chapter 17. Determining the Db2 code level, catalog level, and function level

Before you can activate a Db2 function level, you must ensure that the Db2 subsystem or data sharing group is at the appropriate code level and catalog level.

Procedure

GUI To determine the code level, catalog level, and function level of a Db2 subsystem or data sharing group:

1. Issue a DISPLAY GROUP command.
2. Examine the DISPLAY GROUP output in message DSN7100I.

The DB2 LVL value indicates the code levels of each Db2 subsystem or data sharing member.

Important: Do not attempt to start Db2 at any code level that is lower than the highest ever activated function level, even at the lower star (*) function level. Activate a function level only after you are satisfied that Db2 can continue to run at the required code level.

Examples

The following examples show output from DISPLAY GROUP commands in Db2 13:

-DB2A DISPLAY GROUP DETAIL

Example: Data sharing group with coexisting Db2 13 and Db2 12 members

In the following example, two members are migrated to Db2 13 and ready for the activation of new function in Db2 13. However, the code level of member DB2B indicates that it is not migrated to Db2 13, which means the group is not ready for activation of function level 500. The DB2 LVL value for DB2B is 121510, which indicates that all Db2 12 function levels are activated.

```
-DISPLAY GROUP DETAIL
DSN7100I  -DB2A DSN7GCMDB
*** BEGIN DISPLAY OF GROUP(DSN7G ) CATALOG LEVEL (V13R1M100)
          CURRENT FUNCTION LEVEL (V13R1M100)
          HIGHEST ACTIVATED FUNCTION LEVEL (V13R1M100)
          HIGHEST POSSIBLE FUNCTION LEVEL (V13R1M100)
          PROTOCOL LEVEL (2)
          GROUP ATTACH NAME (DSNG)

-----
DB2      SUB      DB2      SYSTEM      IRLM
MEMBER   ID   SYS  CMDPREF  ELIGIBLE  LVL    NAME    SUBSYS  IRLMPROC
-----
DB2A     1  DB2A  -DB2A    ACTIVE    131501  MVSA     DJ2A    DB2AIRLM
DB2B     2  DB2B  -DB2B    ACTIVE    121510  MVSB     DJ2B    DB2BIRLM
DB2C     3  DB2C  -DB2C    ACTIVE    131501  MVSC     DJ2C    DB2CIRLM
-----

MIGRATION READINESS REPORT
-----
DB2      CODE      SPE      MIGRATION
MEMBER   LEVEL     APAR    ELIGIBLE
-----
DB2A     V13R1M501    .....  N/A
DB2B     V12R1M501    PH37108  YES
DB2B     V12R1M501    PH37108  YES
-----

MIGRATION READINESS STATUS: GROUP MIGRATION IN PROGRESS TO DB2 13
REASON:  ALL REMAINING MEMBERS ARE READY FOR DB2 13

-----
DISPLAY SUBGROUP ATTACH INFORMATION FOR GROUP ATTACH DSNG
-----

SCA  STRUCTURE SIZE:  12288 KB, STATUS= AC,   SCA IN USE:    8 %
LOCK1 STRUCTURE SIZE:  12288 KB
NUMBER LOCK ENTRIES:   1048576
```

```

NUMBER LIST ENTRIES:      23073, LIST ENTRIES IN USE:      7
SPT01 INLINE LENGTH:      32138
*** END DISPLAY OF GROUP(DSNCAT )
DSN9022I  -DB2C DSN7GCMD 'DISPLAY GROUP ' NORMAL COMPLETION

```

Example: Running a DISPLAY GROUP command run on a Db2 12 member during coexistence with Db2 13

The following example output shows the result of running the following command on a Db2 12 member in the data sharing group from the previous example:

```
-DB2B DISPLAY GROUP DETAIL
```

Because the command was run on a Db2 12 member, the highest possible function level indicates V12R1M510, which is the highest level that you can activate by issuing an ACTIVATE command. However, the current function level indicates V13R1M100 because at least one member has been migrated to Db2 13.

```

-DISPLAY GROUP DETAIL
DSN7100I  -DB2B DSN7GCMD
*** BEGIN DISPLAY OF GROUP(DSNCAT ) CATALOG LEVEL(V13R1M100)
          CURRENT FUNCTION LEVEL(V13R1M100)
          HIGHEST ACTIVATED FUNCTION LEVEL(V13R1M100)
          HIGHEST POSSIBLE FUNCTION LEVEL(V12R1M510)
          PROTOCOL LEVEL(2)
          GROUP ATTACH NAME(DSNG)
-----
DB2      SUB      DB2      SYSTEM      IRLM
MEMBER   ID  SYS  CMDPREF  STATUS   LVL    NAME      SUBSYS  IRLMPROC
-----
DB2A     1  DB2A  -DB2A   ACTIVE   131501  MVSA      DJ2A    DB2AIRLM
DB2B     2  DB2B  -DB2B   ACTIVE   121510  MVSB      DJ2B    DB2BIRLM
DB2C     3  DB2C  -DB2C   ACTIVE   131501  MVSC      DJ2C    DB2CIRLM
-----
MIGRATION READINESS REPORT
-----
DB2      CODE      SPE      MIGRATION
MEMBER   LEVEL     APAR     ELIGIBLE
-----
DB2A     V13R1M501   .....  N/A
DB2B     V12R1M510  PH37108  YES
DB2B     V13R1M501   .....  N/A
-----
MIGRATION READINESS STATUS: GROUP MIGRATION IN PROGRESS TO DB2 13
REASON: ALL REMAINING MEMBERS ARE READY FOR DB2 13
-----
DISPLAY SUBGROUP ATTACH INFORMATION FOR GROUP ATTACH DSNG
-----
SCA  STRUCTURE SIZE:      12288 KB, STATUS= AC,   SCA IN USE:      8 %
LOCK1 STRUCTURE SIZE:      12288 KB
NUMBER LOCK ENTRIES:      1048576
NUMBER LIST ENTRIES:      23073, LIST ENTRIES IN USE:      7
SPT01 INLINE LENGTH:      32138
*** END DISPLAY OF GROUP(DSNCAT )
DSN9022I  -DB2B DSN7GCMD 'DISPLAY GROUP ' NORMAL COMPLETION

```

Example: Data sharing group with all active members migrated to Db2 13 code, before the activation of function level 500

The following DISPLAY GROUP output illustrates a data sharing group with all active members migrated to Db2 13 and ready for the activation of function level 500.

```

DSN7100I  -DB2B DSN7GCMD
*** BEGIN DISPLAY OF GROUP(DSNCAT ) CATALOG LEVEL(V13R1M100)
          CURRENT FUNCTION LEVEL(V13R1M100)
          HIGHEST ACTIVATED FUNCTION LEVEL(V13R1M100)
          HIGHEST POSSIBLE FUNCTION LEVEL(V13R1M500)
          PROTOCOL LEVEL(2)
          GROUP ATTACH NAME(DSNG)
-----
DB2      SUB      DB2      SYSTEM      IRLM
MEMBER   ID  SYS  CMDPREF  STATUS   LVL    NAME      SUBSYS  IRLMPROC
-----

```

```

DB2A      1 DB2A -DB2A      ACTIVE  131501 MVSA      DJ2A  DB2AIRLM
DB2B      2 DB2B -DB2B      ACTIVE  131501 MVSB      DJ2B  DB2BIRLM
DB2C      3 DB2C -DB2C      ACTIVE  131501 MVSC      DJ2C  DB2CIRLM
-----
MIGRATION READINESS REPORT
-----
DB2      CODE      SPE      MIGRATION
MEMBER   LEVEL     APAR     ELIGIBLE
-----
DB2A     V13R1M501  .....  NO
DB2B     V13R1M501  .....  NO
DB2C     V13R1M501  .....  NO
-----
MIGRATION READINESS STATUS: GROUP IS NOT READY FOR DB2 NEXT
REASON: NO NEXT RELEASE OF DB2 IS CURRENTLY AVAILABLE
-----
DISPLAY SUBGROUP ATTACH INFORMATION FOR GROUP ATTACH DSNG
-----
SCA  STRUCTURE SIZE:  12288 KB, STATUS= AC,   SCA IN USE:    9 %
LOCK1 STRUCTURE SIZE: 12288 KB
NUMBER LOCK ENTRIES:  1048576
NUMBER LIST ENTRIES:  23073, LIST ENTRIES IN USE:    17
SPT01 INLINE LENGTH:  32138
*** END DISPLAY OF GROUP(DSNCAT )
DSN9022I -DB2B DSN7GCMD 'DISPLAY GROUP ' NORMAL COMPLETION

```

Example: Data sharing group with all active members migrated to Db2 13 after the activation of function level 500

The following DISPLAY GROUP output illustrates a data sharing group with function level 500 activated.

```

DSN7100I -DB2A DSN7GCMD
*** BEGIN DISPLAY OF GROUP(DSNCAT ) CATALOG LEVEL(V13R1M500)
          CURRENT FUNCTION LEVEL(V13R1M500)
          HIGHEST ACTIVATED FUNCTION LEVEL(V13R1M500)
          HIGHEST POSSIBLE FUNCTION LEVEL(V13R1M500)
          PROTOCOL LEVEL(2)
          GROUP ATTACH NAME(DSNG)
-----
DB2      SUB      DB2      SYSTEM      IRLM
MEMBER   ID  SYS  CMDPREF  STATUS  LVL   NAME      SUBSYS  IRLMPROC
-----
DB2A     1 DB2A -DB2A      ACTIVE  131501 MVSA      DJ2A  DB2AIRLM
DB2B     2 DB2B -DB2B      ACTIVE  131501 MVSB      DJ2B  DB2BIRLM
DB2C     3 DB2C -DB2C      ACTIVE  131501 MVSC      DJ2C  DB2CIRLM
-----
MIGRATION READINESS REPORT
-----
DB2      CODE      SPE      MIGRATION
MEMBER   LEVEL     APAR     ELIGIBLE
-----
DB2A     V13R1M501  .....  NO
DB2B     V13R1M501  .....  NO
DB2B     V13R1M501  .....  NO
-----
MIGRATION READINESS STATUS: GROUP IS NOT READY FOR DB2 NEXT
REASON: NO NEXT RELEASE OF DB2 IS CURRENTLY AVAILABLE
-----
DISPLAY SUBGROUP ATTACH INFORMATION FOR GROUP ATTACH DSNG
-----
SCA  STRUCTURE SIZE:  12288 KB, STATUS= AC,   SCA IN USE:    9 %
LOCK1 STRUCTURE SIZE: 12288 KB
NUMBER LOCK ENTRIES:  1048576
NUMBER LIST ENTRIES:  23073, LIST ENTRIES IN USE:    0
SPT01 INLINE LENGTH:  32138
*** END DISPLAY OF GROUP(DSNCAT )
DSN9022I -DB2A DSN7GCMD 'DISPLAY GROUP ' NORMAL COMPLETION

```

Example: Data sharing group with an encryption key label assigned to all members

If the subsystem parameter ENCRYPTION_KEYLABEL is specified for the members of a data sharing group, issue the following command to display the key label:

```
-DISPLAY GROUP DETAIL
```

The output is similar to the following:

```
DSN7100I -DB2C DSN7GCMDB
*** BEGIN DISPLAY OF GROUP(DSNCAT ) CATALOG LEVEL(V13R1M500)
      CURRENT FUNCTION LEVEL(V13R1M500)
      HIGHEST ACTIVATED FUNCTION LEVEL(V13R1M500)
      HIGHEST POSSIBLE FUNCTION LEVEL(V13R1M500)
      PROTOCOL LEVEL(2)
      GROUP ATTACH NAME(DSNG)
      ENCRYPTION KEY LABEL (SYSTEM.KEY01)
```

What to do next

1. If necessary, apply maintenance to the Db2 subsystem or data sharing group members for the code level required by the target function level, and repeat this task.
2. Activate the target function level as described in [Chapter 20, “Activating Db2 13 function levels,” on page 147](#).

Related concepts

Function levels and related levels in Db2 13

Enhancements to Db2 13 are enabled for use when you activate function levels. Before you can activate a function level, the Db2 subsystem, or all Db2 data sharing members, must be at a sufficient code level, and the Db2 catalog must be updated to the required Db2 catalog level.

Related reference

[-DISPLAY GROUP \(Db2\) \(Db2 Commands\)](#)

Related information

[DSN7100I \(Db2 Messages\)](#)

Chapter 18. Testing Db2 function level activation


Before you activate a Db2 function level, you can optionally test whether the Db2 subsystem or data sharing group is ready for activation of the target function level.

Before you begin

Important: When you check the readiness of your Db2 environment for a function level, be careful to specify the TEST option with the ACTIVATE command. After any successful completion of the ACTIVATE command without TEST, Db2 must remain at the higher code level. That is, you cannot remove any PTFs that the code level requires, even at a lower star (*) function level. You can also use the DISPLAY GROUP command to determine the highest function level that your Db2 environment supports, without risk of inadvertent function level activation. For more information, see [Chapter 17, “Determining the Db2 code level, catalog level, and function level,”](#) on page 139.

Procedure

To test activation of a Db2 function level, complete the following steps:

1.  Issue an ACTIVATE command with the TEST option and specify the target Db2 function level to test.

```
-ACTIVATE FUNCTION LEVEL (V13R1M500) TEST
```

2. Examine the DSNU757I message.

The DSNU757I message indicates whether the group is ready for the specified level. Because TEST is specified, the output includes detailed information about each active member of the data sharing group. In this example, all of the members are at the required code level and catalog level so that function level 500 can be activated.

```
DSNU757I  -DB2A DSNUGCCA
*** BEGIN ACTIVATE FUNCTION LEVEL (V13R1M500)
          GROUP ELIGIBLE FOR FUNCTION LEVEL (V13R1M500)
          CATALOG LEVEL(V13R1M500)
          CURRENT FUNCTION LEVEL(V13R1M100)
          HIGHEST ACTIVATED FUNCTION LEVEL (V13R1M100)
          HIGHEST POSSIBLE FUNCTION LEVEL(V13R1M500)
```

DB2 MEMBER	ID	CURRENT CODE-LEVEL	CAPABLE FUNCTION LEVELS LOWEST	HIGHEST	STATUS
DB2A	1	V13R1M500	V13R1M100	V13R1M500	ELIGIBLE
DB2B	2	V13R1M500	V13R1M100	V13R1M500	ELIGIBLE
DB2C	3	V13R1M500	V13R1M100	V13R1M500	ELIGIBLE

```
DSN9022I  -DB2A DSNZACMD '-ACTIVATE FUNC' NORMAL COMPLETION
```

 **GUI**

What to do next

1. If necessary, apply maintenance to the Db2 subsystem or data sharing group members for the code level required by the target function level, and repeat this task.
2. Activate the target function level as described in [Chapter 20, “Activating Db2 13 function levels,”](#) on page 147.

Related concepts

[Function levels and related levels in Db2 13](#)

Enhancements to Db2 13 are enabled for use when you activate function levels. Before you can activate a function level, the Db2 subsystem, or all Db2 data sharing members, must be at a sufficient code level, and the Db2 catalog must be updated to the required Db2 catalog level.

Related reference

[-ACTIVATE \(Db2\) \(Db2 Commands\)](#)

Related information

[DSNU757I \(Db2 Messages\)](#)

Chapter 19. Identifying applications that are incompatible with catalog updates

Before you activate a Db2 function level, you might need to use the CATMAINT utility to tailor the catalog for the target function level. You can identify applications, activities, and Db2 resources that might be incompatible with catalog migrations and updates and take appropriate actions beforehand to minimize the possibility of a failed catalog update.

Procedure

To identify applications that might interfere with a catalog update, take one of the following actions:

- Issue the DISPLAY BLOCKERS command, which displays locks and claims that active threads hold against specified databases.
You can use the optional DETAIL keyword to receive additional report information about each lock or claim.
- Invoke the BLOCKING_THREADS built-in function, which returns a table that contains one row for each lock or claim that threads hold against specified databases.

What to do next

If incompatible applications exist, take one of the following actions:

- Terminate any active threads that hold locks or claims against the specified databases. For more information, see [Canceling threads \(Db2 Administration Guide\)](#).
- Run the CATMAINT utility during a time window when the incompatible applications are not running.
- Schedule a planned outage for running the CATMAINT utility.

Related reference

[-DISPLAY BLOCKERS \(Db2\) \(Db2 Commands\)](#)

[BLOCKING_THREADS \(Db2 SQL\)](#)

Chapter 20. Activating Db2 13 function levels

You control the activation and adoption of new features in Db2 13 by activating function levels and specifying the application compatibility level. You can also continue to apply corrective and preventative service without adopting new feature function.

Before you begin

Ensure that no incompatible applications will interfere with the catalog update. For details, see [Chapter 19, “Identifying applications that are incompatible with catalog updates,”](#) on page 145.

Determine the function level to activate. In most cases, you can activate a higher function level without separately activating each lower function level above the currently activated function level. However, activating a higher function level also results in the activation of all lower function levels. Before activating a function level, familiarize yourself with the new capabilities and changes that all lower function levels introduce:

- [Chapter 3, “Db2 13 function levels,”](#) on page 55
- [Chapter 4, “Incompatible changes in Db2 13,”](#) on page 89
-
- [Part 2, “Changes to plan for in Db2 13,”](#) on page 87

About this task

The ACTIVATE command controls the activation of new function in Db2. You can tailor jobs for updating the Db2 catalog and activating Db2 function levels by running the Db2 installation CLIST.

Tip: You can also use z/OSMF to automate running the jobs for this task. For more information, see [Chapter 21, “Activating Db2 function levels by using z/OSMF,”](#) on page 151.

Procedure

To activate capabilities and enhancements that are introduced by Db2 function levels, complete the following steps:

1. If Db2 13 is at function level 100 or 500, follow the steps in [Activating Db2 13 function level 500 or higher \(Db2 Installation and Migration\)](#).
2. Issue a DISPLAY GROUP command to check that the code level of the Db2 subsystem or each data sharing group member supports your target function level.
In the DSN7100I message, the DB2 LVL column indicates the code level. For more information and examples, see [Chapter 17, “Determining the Db2 code level, catalog level, and function level,”](#) on page 139.
3. If necessary, apply maintenance, such as PTFs and RSUs, to bring your Db2 subsystem or data sharing group members up to the required code level for your target function level.
4. Generate tailored JCL jobs for the CATMAINT and function level activation steps. You can use the DSNTIJBC batch job or the Db2 installation CLIST.

Tip: You can avoid working through the Db2 installation CLIST panels in interactive mode by running a batch job with valid input files to generate the required JCL jobs and input files with a background process. See [Generating tailored Db2 migration or function level activation jobs in the background \(Db2 Installation and Migration\)](#).

To generate the required JCL jobs and input files with a background process, complete the following steps:

- a. Customize the DSNTIDOA parameter override file by following that instructions in the file.

- b. Customize the DSNTIJBC job. For example, if *prefix*.SDSNSAMP(DSNTIDOA) is the customized parameter override file, you can specify the following values in the IPSTART command in DSNTIJBC.

```
ISPSTART CMD(%DSNTINSB +  
  OVERPARM(prefix.SDSNSAMP(DSNTIDOA)) +  
  ) BREDIMAX(1)
```

- c. If you use Db2 Value Unit Edition, you must also provide the data set name of the DSNTIDVU parameter override file in the IPSTART command in the DSNTIJBC job, as shown in the following example, where *prefix*.SDSNSAMP(DSNTIDVU) is the customized OTC LICENSE file.

```
ISPSTART CMD(%DSNTINSB +  
  OVERPARM(<prefix>.SDSNSAMP(DSNTIDOA)) +  
  OTCLPARM(<prefix>.SDSNSAMP(DSNTIDVU)) +  
  ) BREDIMAX(1)
```

- d. Submit the customized DSNTIJBC job.

To generate the required JCL jobs and input files with the Db2 installation CLIST in interactive mode, complete the following steps:

- a. On panel DSNTIPA1, specify values in the ACTIVATE, INPUT MEMBER, and OUTPUT MEMBER fields.
 - In the INSTALL TYPE field, specify ACTIVATE.
 - In the INPUT MEMBER field, specify the name of the CLIST output member that you created when you installed or migrated to Db2 13, or most recently activated a Db2 13 function level.
 - In the OUTPUT MEMBER field, specify a new member name, to save your changes for future use.
 - b. On panel DSNTIPT, verify the SAMPLE LIBRARY field value, which is the name of the output data set that is to be created. An asterisk appears at the far left of this field if the data set already exists. If the data set already exists, the CLIST replaces the members that it customizes for activation of the new function level.
 - c. On panel DSNTIP00, specify V13R1M5nn in the TARGET FUNCTION LEVEL field, where 5nn is the function level that you want to activate. The format is VvvRrMmmm, where vv is the version, r is the release, and mmm is the modification level. The value is used in the ACTIVATE command in the DSNTIJAF job and in the CATMAINT utility control statement in the DSNTIJTC job.
 - d. If all Db2 applications can be bound and run at the target function level, modify the APPLCOMPAT and SQLLEVEL subsystem parameter settings. Otherwise, leave these fields unchanged.
 - e. Proceed through the remaining panels, and wait for the CLIST to tailor the jobs for the activation process. The output data set contains the tailored jobs for the activation process.
5. Run the DSNTIJIC job to take an image copy of the Db2 catalog and directory.
6. Run the tailored DSNTIJTC job to run the CATMAINT utility with LEVEL V13R1M5nn to update the catalog to the appropriate catalog level.

You can use a single CATMAINT job that specifies the target function level. If the target function level requires multiple catalog level updates, the CATMAINT job processes each update in sequential order. If a later update in the sequence fails, the previous successful updates do not roll back, and the catalog level remains at the highest level reached. If that occurs, you can correct the reason for the failure and resubmit the same CATMAINT job.

Important: Do not attempt to start Db2 at a lower code level after any part of the CATMAINT job for a higher function level completes. Run the CATMAINT job only after you are satisfied that Db2 can continue to run at the necessary code level. The code to tolerate catalog changes is contained in the code level that delivers the CATMAINT job.

When the catalog level change completes, Db2 records the change in the SYSIBM.SYSLEVELUDPATES catalog table and issues message DSNB014I to the console.

7. Optional: Test the Db2 code level and catalog level for readiness to activate of the function level.

For more information and examples, see [Chapter 18, “Testing Db2 function level activation,”](#) on page 143.

8. Run the tailored DSNTIJAF job, or issue the following ACTIVATE command, with V13R1M5nn changed to function level that you want to activate.

```
-ACTIVATE FUNCTION LEVEL (V13R1M5nn)
```

Important: Do not attempt to start Db2 at any code level that is lower than the highest ever activated function level, even at the lower star (*) function level. Activate a function level only after you are satisfied that Db2 can continue to run at the required code level.

If the command completes successfully, Db2 issues message DSN9022I. Message DSNU757I indicates the result of the activate command.

GUI For example, the following message indicates successful activation:

```
DSNU757I  -DB2A DSNUGCCA
*** BEGIN ACTIVATE FUNCTION LEVEL (V13R1M500)
          FUNCTION LEVEL (V13R1M500) SUCCESSFULLY ACTIVATED
          CATALOG LEVEL (V13R1M100)
          CURRENT FUNCTION LEVEL (V13R1M500)
          HIGHEST ACTIVATED FUNCTION LEVEL (V13R1M100)
          HIGHEST POSSIBLE FUNCTION LEVEL (V13R1M500)
DSN9022I  -DB2A DSNZACMD '-ACTIVATE FUNC' NORMAL COMPLETION
```

When the function level change completes, Db2 records the change in the SYSIBM.SYSLEVELUDPATES catalog table and issues message DSN9014I to the console.

GUI

More actions are required after the ACTIVATE command completes successfully before most types of new capabilities and enhancements in the function level can be used. For example, new SQL capabilities require that applications use the appropriate application compatibility level, and optimization enhancements apply only after full prepare of the SQL statements.

9. After you are ready for applications to use the new capabilities in the function level, rebind them at the corresponding application compatibility level. For more information, see [Chapter 23, “Controlling the Db2 application compatibility level,”](#) on page 155.

Tip: Proceed with the following steps only if all Db2 applications can be bound and run at the target function level.

Optionally, when you are ready for all applications to use the new capabilities of the target function level, you can run the following jobs:

- a. Run DSNTIJUZ to modify the subsystem parameter module with the APPLCOMPAT value that was specified on panel DSNTIP00.
- b. Run DSNTIJOZ job to issue SET SYSPARM command to bring the APPLCOMPAT subsystem parameter changes online.
- c. Run DSNTIJUA job to modify the Db2 data-only application defaults module with the SQLLEVEL value that was specified on panel DSNTIP00.

What to do next

Complete any of the following actions:

- If the new function level includes optimization enhancements, Db2 must process a full prepare before any SQL statements can benefit. Whether a full prepare occurs depends on the statement type:
 - For static SQL statements, after bind or rebind of the package.
 - For non-stabilized dynamic SQL statements, immediately, unless the statement is in the dynamic statement cache.
 - For stabilized dynamic SQL statements, after invalidation, free, or changed application compatibility level.

- If you did not run the jobs to update the APPLCOMPAT and SQLLEVEL subsystem parameters, resolve any application incompatibilities and increase the application compatibility level of your applications after you are satisfied that Db2 is stable at the target function level, as described in [Chapter 23, “Controlling the Db2 application compatibility level,”](#) on page 155.
- If you encounter regressions or other problems when you activate a Db2 13 function level, minimize the impact to your applications while you resolve the problems by following the general approaches described in [Chapter 24, “Responding to problems after function level activation,”](#) on page 157.
- You might need to update COPY, RECOVER, and REBUILD INDEX utility jobs to add new or remove obsolete Db2 catalog objects. When a utility encounters the following situations for Db2 catalog objects, it issues message DSNU1530I:
 - The object is no longer used or no longer exists.
 - The object is created in a higher catalog level or function level.

For best results, make these updates the utility jobs after you activate the function level that the changes support. For more information about the catalog changes in Db2 13, see [Chapter 9, “Catalog changes in Db2 13,”](#) on page 105.

Related concepts

[Function levels and related levels in Db2 13](#)

Enhancements to Db2 13 are enabled for use when you activate function levels. Before you can activate a function level, the Db2 subsystem, or all Db2 data sharing members, must be at a sufficient code level, and the Db2 catalog must be updated to the required Db2 catalog level.

[Application compatibility levels in Db2 \(Db2 Application programming and SQL\)](#)

Related reference

[Db2 13 function levels](#)

New Db2 capabilities and enhancements are continuously delivered in a single maintenance stream as the code becomes ready. You can activate the new capabilities in a data sharing group or Db2 subsystem after a function level is delivered. A *function level* corresponds to a single PTF that enables the activation of a specific set of enhancements that shipped in previous prerequisite or co-requisite PTFs. The activation of a function level results in the activation of all lower function levels.

[-ACTIVATE \(Db2\) \(Db2 Commands\)](#)

[CATMAINT \(Db2 Utilities\)](#)

Related information

[DSNU757I \(Db2 Messages\)](#)

[DSNG014I \(Db2 Messages\)](#)

Chapter 21. Activating Db2 function levels by using z/OSMF

You can use z/OSMF workflows to automate the running of jobs for activating new capabilities in Db2 function levels.

Before you begin

Important: Before you activate function level 500 or higher for the first time, read [Activating Db2 13 function level 500 or higher \(Db2 Installation and Migration\)](#).

Ensure that no incompatible applications will interfere with the catalog update. For details, see [Chapter 19, “Identifying applications that are incompatible with catalog updates,”](#) on page 145.

Determine the function level to activate. In most cases, you can activate a higher function level without separately activating each lower function level above the currently activated function level. However, activating a higher function level also results in the activation of all lower function levels. Before activating a function level, familiarize yourself with the new capabilities and changes that all lower function levels introduce:

- [Chapter 3, “Db2 13 function levels,”](#) on page 55
- [Chapter 4, “Incompatible changes in Db2 13,”](#) on page 89
-
- [Part 2, “Changes to plan for in Db2 13,”](#) on page 87

About this task

You can run the jobs for activating new capabilities in Db2 function levels by using z/OSMF workflows.

Procedure

To activate a Db2 function level by using z/OSMF, complete the following steps:

1. Issue a DISPLAY GROUP command to check that the code level of the Db2 subsystem or each data sharing group member supports your target function level.
In the DSN7100I message, the DB2 LVL column indicates the code level. For more information and examples, see [Chapter 17, “Determining the Db2 code level, catalog level, and function level,”](#) on page 139.
2. If necessary, apply maintenance, such as PTFs and RSUs, to bring your Db2 subsystem or data sharing group members up to the required code level for your target function level.
3. Run the installation CLIST, as described in [Generating tailored Db2 13 installation, migration, or function level activation jobs \(Db2 Installation and Migration\)](#).
 - a) On panel DSNTIPA1, specify YES in the USE Z/OSMF WORKFLOW field.
 - b) On panel DSNTIPA1, specify ACTIVATE in the INSTALL TYPE field.
 - c) On panel DSNITP00, specify the target function level in the TARGET FUNCTION LEVEL field.
The format is *VvvRrMmmm*, where *vv* is the version, *r* is the release, and *mmm* is the modification level.
The value is used in the ACTIVATE command in the DSNTIJAF job and the CATMAINT utility control statement in the DSNTIJTC job.
4. In z/OSMF, run the DSNTIWAF workflow.
The DSNTIWAF workflow always runs the DSNTIJAF job to run an ACTIVATE command for the target function level. It runs various other jobs depending on the specific situation.

Related tasks

[Automating Db2 migration by using z/OS Management Facility \(Db2 Installation and Migration\)](#)

Related reference

[-ACTIVATE \(Db2\) \(Db2 Commands\)](#)

[CATMAINT \(Db2 Utilities\)](#)

Job DSNTIJOZ: bring Db2 subsystem parameter changes online

Job DSNTIJOZ runs a SET SYSPARM command to bring subsystem parameter changes online.

The z/OSMF workflows for migrating to Db2 13 and activating Db2 function levels use this job.

Related tasks

[Activating Db2 function levels by using z/OSMF](#)

You can use z/OSMF workflows to automate the running of jobs for activating new capabilities in Db2 function levels.

[Automating Db2 migration by using z/OS Management Facility \(Db2 Installation and Migration\)](#)

Related reference

[-SET SYSPARM \(Db2\) \(Db2 Commands\)](#)

Chapter 22. Updating Db2 initialization parameters for function level activation

After you activate a function level, you can enable applications to begin using new capabilities by default by updating the APPLCOMPAT subsystem parameter and the SQLLEVEL value in the application defaults module.

About this task

Tip: Do not raise the default application compatibility level of the Db2 subsystem immediately after migrating or activating a new function level. Instead, wait until applications have been verified to work correctly at the higher function level, and any incompatibilities have been resolved. For details, see [Enabling default application compatibility with function level 500 or higher \(Db2 Application programming and SQL\)](#).

Procedure

To update initialization parameters for function level activation, complete the following steps:

1. Run job DSNTIJUZ.

Job DSNTIJUZ defines the Db2 data-only subsystem parameter module (DSNZPxxx), which consists of the expansion of the following macros: DSN6ARVP, DSN6FAC, DSN6GRP, DSN6LOGP, DSN6SPRM, and DSN6SYSP.

You might need to make the following adjustments before running the job:

- If you added a STEPLIB DD statement to the Db2 start procedures ahead of *prefix*.SDSNEXIT and *prefix*.SDSNLOAD, you can move the SYSLMOD output to that library.

Important: If the DSNTIJUZ job ends with return code 4, check that any warning messages can be safely ignored before continuing to the next step.

When DSNTIJUZ completes, the DSNTINST CLIST performs calculations by using the values that you specified for some of the parameter values that you entered on the panels. These calculations appear in the macro descriptions.

For more information, see [Job DSNTIJUZ: define the Db2 data-only subsystem parameter module \(Db2 Installation and Migration\)](#).

2. Run job DSNTIJUA.

Job DSNTIJUA defines the Db2 data-only application defaults module.

You might need to make the following adjustments before running the job:

- If you added a STEPLIB DD statement to the Db2 start procedures ahead of *prefix*.SDSNEXIT and *prefix*.SDSNLOAD, you can move the SYSLMOD output to that library.

For more information, see [Job DSNTIJUA: define data-only application defaults module \(Db2 Installation and Migration\)](#).

Chapter 23. Controlling the Db2 application compatibility level

You can use the *application compatibility level* of your applications to control the adoption of new capabilities and enhancements, and the impact of incompatible changes. That is, you can separate the Db2 13 migration and function level activation processes from application updates to adopt new features and resolve incompatible changes.

Before you begin

Activate the function level that introduces the new SQL capabilities that your applications will use. For details, see [Part 3, “Adopting new capabilities in Db2 13 continuous delivery,” on page 133.](#)

About this task

You can change the application compatibility level for each application when you are ready for it to run with the features and behavior of a higher Db2 version or function level. The application compatibility level applies to all SQL statements, including data definition statements.

After function level 500 or higher is activated, you can continue run each application with the features and behavior of an earlier Db2 versions or a lower Db2 13 function levels.

Procedure

To control the adoption of new SQL capabilities, and the impact of incompatible changes on your Db2 applications, and objects such as routines and triggers, use the following process:

1. After the activation of a new function level (including function level 500 after migration to Db2 13), continue to run your applications at the same application compatibility level until you are satisfied that your Db2 13 environment is stable at the new function level.
2. Rebind the packages for SPUFI and the productivity-aid sample programs (such as DSNTDP2, DSNTDP4, DSNTIAD, and DSNTIAUL) at the higher application compatibility level, so that database administrators can begin using the new SQL data definition capabilities.
For more information about preparing the sample programs, see [Db2 productivity-aid sample programs \(Db2 Application programming and SQL\)](#).
3. Identify the highest priority applications for the use of new capabilities. Then, identify and resolve any incompatibilities, as described in [Managing application incompatibilities \(Db2 Application programming and SQL\)](#).
4. Bind or rebind your high-priority applications at the higher application compatibility level.

For best results, apply the following approaches when you complete this step:

- Rebind packages for static SQL applications. Specify use of the PLANMGMT bind option so that you can revert to a previous package copy if a regression occurs.
- Rebind packages for dynamic SQL applications. The application compatibility level is also among the matching criteria for both cached and stabilized dynamic statements. When it changes, cached dynamic statements exit the cache and require a full prepare at the next execution, and stabilized dynamic SQL statements are no longer stabilized and subject to full prepare and access path change. It is best to re-stabilize such statements only after you are satisfied that no access path regression has occurred.
- Rebind distributed packages in separate collections and switch the applications to using the new collections.

Tip: Extra program preparation steps might be required to increase the application compatibility level for applications that use data server clients or drivers to access Db2 for z/OS. For more information,

see [Setting application compatibility levels for data server clients and drivers \(Db2 Application programming and SQL\)](#).

5. Repeat the two preceding steps for any applications, and any objects such as routines or triggers, that require the use of new SQL capabilities.
6. After incompatible changes are resolved for most applications, rebind any remaining applications that must continue to run compatibly with the lower level, and explicitly specify the lower application compatibility level.

What to do next

After all applications are ready to run at a higher application compatibility level or explicitly bound at a lower level, you can increase the APPLCOMPAT subsystem parameter value to bind packages at a higher application compatibility level by default.

The APPLCOMPAT subsystem parameter specifies the default value to use when the APPLCOMPAT bind option is not specified in a BIND command, or the APPLCOMPAT value is not specified or stored in the Db2 catalog for a REBIND command. Its value does not prevent specific applications from running at higher application compatibility levels. For more information, see [APPL COMPAT LEVEL field \(APPLCOMPAT subsystem parameter\) \(Db2 Installation and Migration\)](#).

For detailed instructions, see [Enabling default application compatibility with function level 500 or higher \(Db2 Application programming and SQL\)](#).

Related concepts

[Application compatibility levels in Db2 \(Db2 Application programming and SQL\)](#)

[Function levels and related levels in Db2 13](#)

Enhancements to Db2 13 are enabled for use when you activate function levels. Before you can activate a function level, the Db2 subsystem, or all Db2 data sharing members, must be at a sufficient code level, and the Db2 catalog must be updated to the required Db2 catalog level.

Related tasks

[Responding to problems after function level activation](#)

If you encounter regression or other problems after the activation of a new Db2 13 function level, you can take certain actions to minimize the impact to your applications while you resolve the underlying problems.

[Setting application compatibility levels for data server clients and drivers \(Db2 Application programming and SQL\)](#)

Related reference

[APPL COMPAT LEVEL field \(APPLCOMPAT subsystem parameter\) \(Db2 Installation and Migration\)](#)

Chapter 24. Responding to problems after function level activation

If you encounter regression or other problems after the activation of a new Db2 13 function level, you can take certain actions to minimize the impact to your applications while you resolve the underlying problems.

About this task

You might sometimes activate a lower function level, called a *star (*) function level* after you activate a higher function level. Any function level lower than the highest function level that was ever activated is always a star (*) function level.

The output from DISPLAY GROUP and ACTIVATE commands identify star (*) function levels by the function level identifier followed by an asterisk, hence the name. For example, assume that you activate function level 500 after function level 501 was previously activated. V13R1M500* in the message output indicates that the Db2 data sharing group or subsystem is at function level 500* (you might say, "function level 500 star").

Important: Activating a lower star (*) function level does not enable you to remove maintenance and start Db2 at any code level lower than the catalog level or highest ever activated function level.

Activating a lower star (*) function level by itself does not immediately disable all use of capabilities at higher function levels. Instead, it provides flexibility for limiting or disabling the use of capabilities at higher function levels, while the problems encountered in higher function levels are resolved.

At a star (*) function level, Db2 continues to tolerate objects, packages, and structures that were created or bound at higher function levels. Also, in any context where the effective application compatibility level remains at the higher level, new SQL capabilities from the higher level can still be used. For packages, they can still be executed, rebound, and automatically bound. However, an explicit bind of such packages succeeds only when the APPLCOMPAT bind option is equivalent to the activated star (*) function level or lower. Similar rules apply for the application compatibility levels of native SQL procedures, compiled SQL scalar functions, and advanced triggers. The result is that applications that use capabilities at a higher function level can continue to do so if they are not related to the reason for activating the lower function level. To stop the use of all SQL capabilities at the higher function level, you must also set all effective application compatibility levels at the lower level.

Important: Do not attempt to start Db2 at any code level that is lower than the highest ever activated function level, even at the lower star (*) function level. Activate a function level only after you are satisfied that Db2 can continue to run at the required code level.

Important: Do not attempt to start Db2 at a lower code level after any part of the CATMAINT job for a higher function level completes. Run the CATMAINT job only after you are satisfied that Db2 can continue to run at the necessary code level. The code to tolerate catalog changes is contained in the code level that delivers the CATMAINT job.

Procedure

Use the following general approaches to minimize the impact of problems at a new function level, while you resolve the underlying issues:

1. If issues occur after you rebind packages at a higher application compatibility level, do not immediately revert to a lower star (*) function level. Instead, use REBIND SWITCH(PREVIOUS) to revert to the previous package.

This option is available only if you used PLANMGMT at the previous bind or rebind of the package.

2. If necessary, issue an ACTIVATE command or tailor and run job DNSTIJAF to activate the lower star (*) function level.

Activating the star (*) function level does not prevent the use of new SQL capabilities. You might continue to run applications not causing or related to the regression or problem at the higher application compatibility level. Such application can continue to use capabilities of the higher function level, unless you bind them at the lower application compatibility level.

3. If you must prevent all use of capabilities at the higher function level, bind all packages at the application compatibility level that corresponds to the star (*) function level.
4. If the default application compatibility level was set at the higher function level than the star (*) function level, reduce the default application compatibility level to that lower level to prevent bind failures.

For instructions for setting the default application compatibility, see [Enabling default application compatibility with function level 500 or higher \(Db2 Application programming and SQL\)](#).

Related tasks

Controlling Db2 application compatibility

You can use the *application compatibility level* of your applications to control the adoption of new capabilities and enhancements, and the impact of incompatible changes. That is, you can separate the Db2 13 migration and function level activation processes from application updates to adopt new features and resolve incompatible changes.

Chapter 25. How Db2 function levels are documented

Generally, Db2 for z/OS documentation assumes that the highest available function level is activated, and that your applications run with the equivalent application compatibility level in effect. However, new and changed content are marked for changes introduced by function levels.

[FL 500](#) Throughout the Db2 13 information, new and changed content for function levels are marked like this paragraph, with a link to the function level that introduced the changes. You can click the link to see the overview page for the function level. If an entire topic is new, you'll find a single function level overview page link near the beginning of the new topic.

In reference information, syntax diagrams always reflect the highest available function level, and syntax diagrams never include internal revision marks. The associated option descriptions also reflect the highest available function level. However, they are marked with revision marks and links to the function level overview page. In cases where reference materials continue to describe the behavior of previous function levels or application compatibility levels, the differences are generally described in the "Notes" section at the end of the topic.

Related tasks

[Adopting new capabilities in Db2 13 continuous delivery](#)

In Db2 13, function levels and application compatibility levels control the adoption of most new capabilities by Db2 subsystems and Db2 applications.

Related reference

[Db2 13 function levels](#)

New Db2 capabilities and enhancements are continuously delivered in a single maintenance stream as the code becomes ready. You can activate the new capabilities in a data sharing group or Db2 subsystem after a function level is delivered. A *function level* corresponds to a single PTF that enables the activation of a specific set of enhancements that shipped in previous prerequisite or co-requisite PTFs. The activation of a function level results in the activation of all lower function levels.

Information resources for Db2 for z/OS and related products

You can find the online product documentation for Db2 13 for z/OS and related products in IBM Documentation.

For all online product documentation for Db2 13 for z/OS, see [IBM Documentation](https://www.ibm.com/docs/en/db2-for-zos/13) (<https://www.ibm.com/docs/en/db2-for-zos/13>).

For other PDF manuals, see PDF format manuals for Db2 13 for z/OS (<https://www.ibm.com/docs/en/db2-for-zos/13?topic=zos-pdf-format-manuals-db2-13>).

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.

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 13 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 13 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 13 for z/OS.

General-use Programming Interface and Associated Guidance Information is identified where it occurs by the following markings:

 General-use Programming Interface and Associated Guidance Information... 

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:

 Product-sensitive Programming Interface and Associated Guidance Information... 

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 Documentation

For definitions of Db2 for z/OS terms, see [Db2 glossary \(Db2 Glossary\)](#).

Index

Numerics

10-byte format
 extended format
 RBA and LRSN [123](#)
 RBA and LRSN [123](#)

C

catalog levels
 determining [139](#)
code levels
 determining [139](#)
commands
 changes [95](#)

D

Db2 13
 overview of new function [3](#)
Db2 commands
 DISPLAY GROUP
 examples [139](#)
Db2function levels
 determining [139](#)
DISPLAY GROUP
 command examples [139](#)

E

enhancements by function level [55](#), [135](#), [147](#)
EXPLAIN tables
 changes [109](#)
 Db2 12 format [109](#)
 encoding [109](#)

F

function level 500 [71](#)
function level activation
 incompatible applications [145](#)
function levels
 determining [139](#)

G

general-use programming information, described [164](#)
GUIP symbols [164](#)

I

IFCID (instrumentation facility component identifier)
 changed [111](#)
 new [111](#)
incompatibilities of releases [89](#)
incompatible applications

incompatible applications (*continued*)
 identifying [145](#)

L

links
 non-IBM Web sites
 [164](#)
LRSN
 extended [123](#)

M

migration
 considerations [89](#)

O

overview
 Db2 13 new function [3](#)

P

product-sensitive programming information, described [164](#)
programming interface information, described [164](#)
PSPI symbols [164](#)

R

RBA
 extended [123](#)
release incompatibilities [89](#)

S

SQL statements
 changes [99](#)

U

utilities
 changes [103](#)



Product Number: 5698-DB2
5698-DBV

GC28-2784-00

