Db2 12 for z/OS

What's New?
Last updated: 2024-04-18

IBM

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.

Contents

About this information

This information provides an executive overview of new function in Db2 12 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 12 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 12 for z/OS, always use IBM® Documentation or download the latest PDF file from PDF format manuals for Db2 12 for z/OS (Db2 for z/OS in IBM Documentation).

Most documentation topics for Db2 12 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 12 migration process and how to activate new capabilities in function levels:

  - Migrating to Db2 12 (Db2 Installation and Migration)
  - Part 1, "What's new in Db2 12," on page 1
  - Part 2, "Adopting new capabilities in Db2 12 continuous delivery," on page 261

- FL 501 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 12. For more information, see Chapter 17, "How Db2 function levels are documented," on page 293.

The availability of new function depends on the type of enhancement, the activated function level, and the application compatibility levels of applications. In the initial Db2 12 release, most new capabilities are enabled only after the activation of function level 500 or higher.

Virtual storage enhancements
Virtual storage enhancements become available at the activation of the function level that introduces them or higher. Activation of function level 100 introduces all virtual storage enhancements in the initial Db2 12 release. That is, activation of function level 500 introduces no virtual storage enhancements.

Subsystem parameters
New subsystem parameter settings are in effect only when the function level that introduced them or a higher function level is activated. Many subsystem parameter changes in the initial Db2 12 release take effect in function level 500. For more information about subsystem parameter changes in Db2 12, see "Subsystem parameter changes in Db2 12" on page 200.

Optimization enhancements
Optimization enhancements become available after the activation of the function level that introduces them or higher, and full prepare of the SQL statements. When 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

Activation of function level 100 introduces all optimization enhancements in the initial Db2 12 release. That is, function level 500 introduces no optimization enhancements.

SQL capabilities

New SQL capabilities become available after the activation of the function level that introduces them or higher, for applications that run at the equivalent application compatibility level or higher. New SQL capabilities in the initial Db2 12 release become available in function level 500 for applications that run at the equivalent application compatibility level or higher. You can continue to run SQL statements compatibly with lower function levels, or previous Db2 releases, including Db2 11 and DB2® 10. For details, see Application compatibility levels in Db2 12 (Db2 Application programming and SQL)

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 rebranded DB2 to Db2, and Db2 for z/OS is the new name of the offering that was previously known as "DB2 for z/OS". For more information, see Revised naming for IBM Db2 family products on IBM z/OS platform. As a result, you might sometimes still see references to the original names, such as "DB2 for z/OS" and "DB2", in different IBM web pages and documents. If the PID, Entitlement Entity, version, modification, and release information match, assume that they refer to the same product.

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 12 for z/OS product documentation (Db2 for z/OS in IBM Documentation)
Related reference
PDF format manuals for Db2 12 for z/OS (Db2 for z/OS in IBM Documentation)

Part 1. What's new in Db2 12

Db2 12 for z/OS takes Db2 to a new level, both extending the core capabilities and empowering the future. Db2 12 extends the core with new enhancements to scalability, reliability, efficiency, security, and availability. Db2 12 also empowers the next wave of applications in the cloud, mobile, and analytics spaces.

Continuous delivery and function levels

Db2 12 introduces *continuous delivery* of new capabilities and enhancements in a single service stream as soon as they are ready. The result is that you can benefit from new capabilities and enhancements without waiting for an entire new release. Some new features and capabilities become available as soon as you apply the PTF for the APAR that introduces them. For a list of these features, see Chapter 1, "New-function APARs for Db2 12," on page 3.

When necessary, *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.

For more information about function levels in Db2 and how to activate them, see:

- Chapter 2, "Db2 12 function levels," on page 69
- Part 2, "Adopting new capabilities in Db2 12 continuous delivery," on page 261
- Exploring IBM Db2 for z/OS Continuous Delivery (IBM Redpaper)

Db2 12 function levels 100 and 500 represent the initial set of new capabilities for Db2 12. In many respects, these function levels are comparable to conversion mode (which corresponds to function level 100) and new-function mode (which corresponds to function level 500) from past releases.

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 12 for z/OS (Db2 for z/OS in IBM Documentation).

Availability of new features and capabilities in Db2 12

Many new capabilities in Db2 12 are introduced in Db2 12 function levels. However, some become available immediately in the Db2 12 *initial release*, or when you apply maintenance.

For migration from Db2 11, most new capabilities become available only after you activate function level 500 or higher.

The availability of new function depends on the type of enhancement, the activated function level, and the application compatibility levels of applications. In the initial Db2 12 release, most new capabilities are enabled only after the activation of function level 500 or higher.

Virtual storage enhancements
Virtual storage enhancements become available at the activation of the function level that introduces them or higher. Activation of function level 100 introduces all virtual storage enhancements in the initial Db2 12 release. That is, activation of function level 500 introduces no virtual storage enhancements.

Subsystem parameters
New subsystem parameter settings are in effect only when the function level that introduced them or a higher function level is activated. Many subsystem parameter changes in the initial Db2 12 release take effect in function level 500. For more information about subsystem parameter changes in Db2 12, see "Subsystem parameter changes in Db2 12" on page 200.

Optimization enhancements

Optimization enhancements become available after the activation of the function level that introduces them or higher, and full prepare of the SQL statements. When 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

Activation of function level 100 introduces all optimization enhancements in the initial Db2 12 release. That is, function level 500 introduces no optimization enhancements.

SQL capabilities

New SQL capabilities become available after the activation of the function level that introduces them or higher, for applications that run at the equivalent application compatibility level or higher. New SQL capabilities in the initial Db2 12 release become available in function level 500 for applications that run at the equivalent application compatibility level or higher. You can continue to run SQL statements compatibly with lower function levels, or previous Db2 releases, including Db2 11 and DB2 10. For details, see Application compatibility levels in Db2 12 (Db2 Application programming and SQL)

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 12 for z/OS Technical Overview (IBM Redbooks)
Db2 12 for z/OS Performance Topics (IBM Redbooks)
Db2 12 for z/OS Optimizer
Exploring IBM Db2 for z/OS Continuous Delivery (IBM Redpaper)

Chapter 1. New-function APARs for Db2 12

Db2 12 introduces the continuous delivery of new capabilities in a single service stream. Many APARs in Db2 12 deliver deactivated functional code to support future function levels. However, some introduce their enhancements immediately, at any Db2 function level.

The following table summarizes the new-function APARs for Db2 12 from newest to oldest, including the APARs that enable activation of Db2 12 function levels. APARs with new-function code for future function levels are listed with the APAR that enables activation of each function level. Only APARs that have PTFs for Db2 12 are included. The following table does not include "bug fixes" or other maintenance APARs.

Tip: Depending on how when and how you order the product code for Db2 12 or Db2 13, you might find that the external changes from any of the following APARs are already built-in when you install a new Db2 environment or migrate to Db2 13. Also, you can expect that external changes from the following APARs with availability dates earlier than 2022-06 are likely to be already built-in when you migrate to Db2 13, regardless of whether you apply the PTFs in Db2 12.

Description	APAR	Date
"Detail information for a page with minimum LRSN that causes a GRECP recovery delay" on page 11	PH54199	2024-01
"STATIME_DDF support" on page 11	PH56228	2023-12
"More granular accounting trace records for distributed applications" on page 12	PH55241	2023-08
"Improved index traverse count information in DISPLAY STATS output" on page 13	PH55255	2023-08
"Latch monitoring and serviceability enhancements" on page 13	PH41756	2023-07
"Migration readiness reported in DISPLAY GROUP command output" on page 13	PH50072	2023-06
"APPLCOMPAT and PLANMGMT support in DSNTIJRT/DSNTRIN for bind or rebind of packages for Db2-supplied routines " on page 15	PH50704	2023-03
"Enhanced SQLDA output for LOBs and distinct types in REXX applications" on page 14	PH52852	2023-04
"APPLCOMPAT and PLANMGMT support in DSNTIJRT/DSNTRIN for bind or rebind of packages for Db2-supplied routines " on page 15	PH50704	2023-03
"Improved default behavior for management of real storage" on page 15	PH47163	2023-02
"Installation and migration changes to support new EXPLAIN columns" on page 16	PH48503	2023-01
"Improved IFCID 0402 reset behavior" on page 16	PH50729	2023-01
"Control the maximum number of concurrent user-defined external scalar functions" on page 16	PH44833	2022-12
"Lock avoidance for singleton SELECT with ISOLATION(CS) and CURRENTDATA(YES)" on page 16	PH49335	2022-12
"Authorization update for CREATE INDEX on DGTT" on page 17	PH48601	2022-11
"Accelerator expression-offload support for LISTAGG and RAND " on page 17	PH48480	2022-11

Description	APAR	Date
"Support for UNI_90 locale for the LOWER, TRANSLATE, and UPPER built-in functions" on page 17	PH47187	2022-09
"DECFLOAT datatype support for PL/I and C with DCLGEN" on page 18	PH47453	2022-08
"Improved DBAT status information in DISPLAY THREAD output" on page 18	PH45504	2022-07
"New serviceability message for LOAD FORCE and REORG FORCE" on page 19	PH44941	2022-06
"New REORG option to minimize the duration of the last log iteration" on page 19	PH33864	2022-05
"New LOAD option to reclaim NPI space" on page 19	PH39194	2022-05
"Improvements to the -START ML and -STOP ML commands " on page 19	PH43479	2022-05
"New statistics trace classes for monitoring distributed user statistics" on page 20	PH40244, PH40243	2022-04
"Improved detection and removal of incomplete connections to Db2" on page 20	PH44270	2022-04
"DCLGEN updated to use COMP-5 declarations for integer data for COBOL applications" on page 20	PH42857	2022-03
"Statistics class 1 includes IFCID 0369" on page 21	PH43916	2022-03
"Controlling application compatibility for productivity-aid sample programs" on page 21	PH41968	2022-02
"LOAD improvements for inline image copies of partitions " on page 21	PH40709	2022-01
"Consistent inline image copies with LOAD REPLACE" on page 22	PH39300	2021-12
"Reduced console messages for FTB" on page 22	PH41751	2021-12
"Support for disabling the filtering of alter or modify log record collection in IFCID 306 trace applications" on page 22	PH40937	2021-12
"Longer password phrase support for connections to Db2 for z/OS" on page 23	PH40443	2021-11
"Improved identification of running log reader tasks from Integrated Synchronization products" on page 23	PH40261	2021-10
"The SYSPROC.ADMIN_COMMAND.DSN stored procedure now supports DCLGEN" on page 23	PH28863	2021-08
"Support for new SUBSTR_COMPATIBILITY subsystem parameter" on page 23	PH36071	2021-08
"IBM Text Search for Db2 for z/OS V5.5.4" on page 24	PH35720	2021-08
"Improved APREUSE bind option functionality" on page 24	PH36728	2021-06
"IBM Db2 for z/OS Developer Extension 1.2" on page 24	None	2021-06
"Support for Redirected Recovery with the RECOVER utility" on page 24	PH35266	2021-06

Description	APAR	Date
"Support for limiting image copy data sets for REORG " on page 25	PI75518	2021-06
"SQL tuning capabilities added to Db2 Accessories Suite 4.2" on page 25	None	2021-06
"Fast index traversal (FTB) support for non-unique indexes" on page 25	PH30978	2021-06
"Improved REORG INDEX utility performance" on page 26	PH25217	2021-05
"RESTSERVICEDEFAULT option for REBIND PACKAGE " on page 26	PH34544	2021-05
"Display information about use of fast index traversal (FTB) with DISPLAY STATS " on page 26	PH34859	2021-05
"Db2 12 function level 510 activation" on page 27 • Prepare for migration to Db2 13	PH33727	2021-04
"REORG utility improvements for SORTNPSI YES" on page 27	PH34403	2021-04
"Increased password size for Db2-supplied stored procedures" on page 27	PH35411	2021-04
"Python support on Db2 for z/OS through ODBC driver with IBM_DB Pyhton support for DB2 for z/OS " on page 27	PH30575	2021-03
"Clean-up phased-out copies for rebind phase-in" on page 28	PH33295	2021-03
"Real-time monitoring of CICS attachment facility calls" on page 28	PH31447	2021-02
"Db2 12 function level 509 activation" on page 28 • "Support for tamper-proof audit policies" on page 72 (PH31180) • "High availability for accelerator-only tables" on page 72 (PH30574) • "Specify a compression algorithm at the table, table space, or partition level" on page 72 (PH31729, PH31730, PH31731)	PH33015	2021-01
"Improved transaction execution times and concurrency for REBIND PACKAGE" on page 28	PH28693	2021-01
"Enhanced DEFAULTIF specifications for the LOAD utility" on page 29	PH30610	2021-01
"LOAD REPLACE or REBUILD INDEX utilities now reset RTS for GETPAGES" on page 29	PH32224	2021-01
"Integrated Accelerator for Z Sort use for GROUP BY or ORDER BY" on page 30	PH31684	2020-12
"Controlling all distributed threads and connections with profile tables" on page 30	PH30780	2020-12
"TEMPLATE utility support for LBI" on page 30	PH30093	2020-12
"Improved status messages for DDF" on page 30	PH30222	2020-11
"Authorized programs can pass eight-byte authorization IDs to Db2" on page 31	PH28017	2020-10
"Support for Redirected Recovery with the RECOVER utility" on page 31	PH27043	2020-10

Description	APAR	Date
"Obsolete subsystem parameters are removed" on page 31	PH28280	2020-10
"Db2 12 function level 508 activation " on page 32 • "Migration of multi-table table spaces to partition-by-growth universal table spaces (UTS)" on page 75 (PH27370, PH27658, PH27659, PH27660, PH27661, PH27662, PH27663) • "FTB support for non-unique indexes" on page 75 (PH30978)	PH29392	2020-10
"INSERT, UPDATE or DELETE statements prepared with the isolation-clause attribute WITH UR are disallowed" on page 32	PH14791	2020-10
"REORG TABLESPACE utility support for IBM Integrated Accelerator for Z Sort feature " on page 33	PH28183	2020-09
"REPAIR DBD TEST and DIAGNOSE allowed on the work file database" on page 33	PH10358	2020-08
"Control of LOAD utility updates to MAXASSIGNEDVAL catalog values for identity columns" on page 33	PH28476	2020-08
"LOAD utility can load data into read-only objects" on page 34	PH26131	2020-08
"MFA authentication support for clients not configured for sysplex WLB" on page 34	PH21341	2020-08
"New option to force the canceling of claimers during LOAD" on page 34	PH24369	2020-08
"CHECK utilities with SHRLEVEL CHANGE reset CHECK-pending status" on page 35	PH25593	2020-08
"LOAD option to define auxiliary objects" on page 35	PH19072	2020-08
"Availability of IBM Db2 for z/OS Developer Extension " on page 35	None	2020-07
"OBJECT_CREATE_FORMAT and UTILITY_OBJECT_CONVERSION subsystem parameters removed" on page 35	PH26317	2020-07
"Message DSNS005I now issued to help explain why storage contraction began" on page 36	PH25940	2020-07
"LOAD can override row change timestamp columns that are defined as GENERATED ALWAYS" on page 36	PH25572	2020-07
"LOAD can presort data" on page 36	PH23105	2020-06
"GETPAGES columns in RTS tables now wrap after reaching the largest supported value" on page 36	PH22838	2020-06
"IBM Db2 AI for z/OS 1.3.0" on page 37	PH15742 PH16111 PH2409 PH25342	2020-06
"LOAD support for dates in 3-byte packed decimal format" on page 37	PH22944	2020-06
"Db2 12 function level 507 activation" on page 37 • "Application granularity for locking limits (NUMLKUS and NUMLKTS)" on page 78 (PH15342)	PH24371	2020-06

Description	APAR	Date
• "Deletion of old statistics when using profiles" on page 78 (PH16345) • "CREATE OR REPLACE for procedures" on page 78 (PH24324) • "Newly supported passthrough-only expressions with IBM Db2 Analytics Accelerator" on page 79 (PH23042)		
"Enable fast index traversal (FTBs) for specific indexes only" on page 37	PH23238	2020-06
"COMPRESS_SPT01 and SPT01_INLINE_LENGTH subsystem parameters removed " on page 38	PH24358	2020-06
"Support client information to be propagated from a Db2 REST call " on page 38	PH24619	2020-06
"Improved reliability and contention relief for online migration" on page 39	PH10826	2020-05
"Estimate space savings for Huffman compression using DSN1COMP " on page 39	PH19242	2020-04
"Support for more granular collection of certain Db2 statistics traces " on page 39	PH19658	2020-04
"Support for deleting remote services and related packages with FREE SERVICE commands" on page 40	PH21058	2020-03
"Subsystem parameter default settings and ranges adjusted for Db2 best practices" on page 40	PH21370	2020-03
"Support for creating artifacts for services that provision Db2 for z/OS data sharing groups" on page 41	PH19751	2020-03
"Add support for UNI_60 locale for the LOWER, TRANSLATE, and UPPER built-in functions" on page 41	PH19888	2020-03
"Serviceability improvements for insert algorithm 2 " on page 41	PH18944	2020-03
"LOAD tolerates invalid PARALLEL specifications " on page 42	PH19073	2020-03
"Automatic rebind operations attempt to reuse the access path from the current package " on page 42	PH15896	2020-02
"Ability to control the number of queued connections waiting for MONITOR THREADS profiles " on page 42	PH12041	2020-02
"Ability to block certain Db2 utilities from updating replicated tables " on page 43	PH14363	2020-01
"Enable SECADM authority to grant BINDAGENT privilege to the binder" on page 43	PH11309	2020-01
"Data set encryption key labels returned by ADMIN_DS_LIST" on page 43	PH12920	2020-01
"Statistics profile options included in utility output " on page 43	PH11423	2020-01
"Validation for default buffer pools in the installation CLIST" on page 44	PH12117	2019-12
"Support for deploying Db2 native REST services" on page 44	PH14688	2019-10
"Db2 12 function level 506 activation" on page 44	PH16829	2019-10

Description	APAR	Date
• "Alternative function names support" on page 81 (PH14712) • "Support for implicitly dropping explicitly created UTS and LOB table spaces" on page 82 (PH14452)		
"IBM Db2 AI for z/OS 1.2.0.2" on page 44	PH14083 PH16093	2019-10
"Ability to limit query acceleration processing to new accelerators" on page 45	PH09995	2019-10
"Ability to control workload balancing of accelerated queries based on accelerator version" on page 45	PH05477	2019-09
"IBM z15: Data privacy for diagnostics for buffer pool data in SVC dumps " on page 45	PH15940	2019-09
"REPAIR utility enhancement to allow write of Db2 diagnostic log record" on page 45	PH11871	2019-09
"Enhancements to insert algorithm 2 " on page 46	PH02052	2019-09
"Console message for inoperative packages" on page 46	PH13550	2019-08
"REORG SHRLEVEL CHANGE is compatible with LOAD RESUME YES SHRLEVEL CHANGE" on page 46	PH11255	2019-08
"Hybrid Transactional Analytical Processing support for accelerated static queries" on page 47	PH14116	2019-08
"New job DSNTIJPM premigration report 24: active log data sets larger than 4 GB" on page 47	PH11514	2019-07
"Support for sysplex group authentication " on page 47	PI94236	2019-07
"Asynchronous cross-invalidation of group buffer pools" on page 48	PH10577	2019-06
"Function level 505 activation" on page 48 • "Rebind phase-in for packages that are being used for execution" on page 85 (PH12186) • "New built-in functions for encryption and decryption with key labels" on page 86 (PH09506) • "Improved support for DECFLOAT columns" on page 86 (PH09797) • "Improved RUNSTATS performance with automatic page sampling by default" on page 86 (PH07220) • "Temporal and archive transparency support for WHEN clause on triggers" on page 86 (PH09794)	PH09191	2019-06
"Loading multiple input data sets in a single job" on page 48	PI96136	2019-06
"Multiple captures on a single proxy for GDPS Continuous Availability with zero data loss " on page 48	PI80787	2019-06
"Creating artifacts for provisioning Db2 subsystems with the CLIST " on page 49	PH09857	2019-06
"Db2 12 Function level 504 activation" on page 49 • "IBM z14 Huffman compression" on page 89 (PH04424)	PH07672	2019-04

Description	APAR	Date
• "Prevent creation of new deprecated objects" on page 89 (PH02873) • "Newly supported built-in functions with IBM Db2 Analytics Accelerator" on page 91 (PH00224) • "New SQL syntax alternatives for special registers and NULL predicates" on page 91 (PH02646)		
"Setting the client application compatibility property is optional when the client is Db2 for Linux, UNIX, and Windows Version 11.1 Fix pack 1 or later" on page 49	PH08482	2019-04
"Allowing the PORT and SECPORT to be the same for static location aliases" on page 50	PH08188	2010-04
"Ability to delete only FlashCopy image copies" on page 50	PH04023	2019-04
"Loading multiple input data sets in a single job" on page 48	PI96136	2019-04
"New job DSNTIJPM premigration report 23: SQL routines created in DB2 9" on page 50	PH07660	2019-04
"Improve RUNSTATS performance by making page sampling the default behavior" on page 50	PH07220	2019-04
"New environment variable to control the MQListener log size" on page 51	PH07230	2019-03
"IBM Db2 AI for z/OS 1.2.0 " on page 51	PH07782	2019-03
"IBM Db2 AI for z/OS1.1.0.1 " on page 51	PH05323	2019-02
"New in-memory resource limit table automatic refresh capability and enhancements to START RLIMIT " on page 51	PH06082	2019-02
"Specify the default padding behavior for REORG TABLESPACE with UNLOAD EXTERNAL or DISCARD processing " on page 52	PH00317	2019-02
"Avoid external sorts for single-column column groups at the utility job level " on page 52	PH03678	2019-02
"Node.js support in Db2 for z/OS with the IBM_DB Node.js driver type " on page 52	PH05953	2019-01
"Db2 exploitation of asynchronous cross-invalidation for coupling facility cache structures" on page 53	PH05193	2018-12
"Improved operational control for Db2 native REST services " on page 53	PI98649	2018-11
"New installation jobs for some DSNTIJMV job tasks" on page 54	PH02971	2018-11
"New alternate format for timestamp values" on page 54	PH03263	2018-11
"Support for tracing native SQL routines" on page 55	PI44721	2018-10
"Function level 503 activation" on page 55 • "Enablement for replication of system-period temporal tables and generated expression columns" on page 95 • "Console message for catalog level or function level change" on page 95 • "Temporal auditing" on page 95 (PI95480)	PH00506	2018-10

Description	APAR	Date
"Allow ROTATE PARTITION for materialized query tables or tables having dependent materialized query tables" on page 55	PH00194	2018-10
"Specify a DDF IPNAME" on page 56	PI99403	2018-10
"Db2 command line processor (CLP) support for user name and credentials for running SQL files" on page 56	PH00335	2018-09
"Specify an HFS or zFS directory for z/OSMF artifacts " on page 57	PI97635	2018-07
"UNLOAD utility uses REGISTER NO by default in Db2 12 " on page 57	PI99075	2018-07
"Suppress SQL warning messages for OPEN and FETCH from SPUFI, DSNTEP2, DSNTEP4, and DSNTIAUL" on page 57	PI79053	2018-06
"Db2 12 Function level 502 activation" on page 58 • "Key label management for z/OS DFSMS data set encryption" on page 98 (PI90288) • "Explicit casting of numeric values to GRAPHIC or VARGRAPHIC" on page 98 (PI88790) • "Db2 12 migration process changes" on page 99	PI95511	2018-05
"User-defined function with MODIFIES SQL DATA from a subselect" on page 58	PI93887	2018-04
"Hybrid Transactional Analytical Processing support for accelerated queries" on page 58	PI91620	2018-03
"LOAD REPLACE SHRLEVEL REFERENCE " on page 58	PI69095	2018-03
"Population of null attributes for objects created before function level V12R1M500" on page 59	PI86123	2018-01
"HFS DRBM support for COBOL and PL/I" on page 59	PI88171	2018-01
"Db2 Native REST services support update" on page 59	PI86868	2018-01
"Automatic remigration binds are removed in release coexistence and fallback " on page 60	PI87675	2018-01
"IBM zHyperLink: database synchronous read I/O exploitation" on page 60	PI82575	2018-01
"LOAD RESUME SUPPORT FOR INLINE COPY" on page 60	PI81724	2017-12
"MQListener automatic restart after Db2 restart " on page 61	PI84698	2017-12
"New console message for remote client information" on page 61	PI89903	2017-12
"DSNTIJSG installation job no longer creates SYSIBM EXPLAIN tables " on page 61	PI86450	2017-12
"CATMAINT utility LEVEL supports function level values" on page 61	PI88058	2017-12
"DSNTIJUZ split into task-specific installation jobs " on page 62	PI85657	2017-11
"Reduced cost for collecting Db2 frequency statistics" on page 62	PI76730	2017-11
"Changes to address problems after Db2 table definition changes " on page 63	PI86880 PI88940	2017-11

Description	APAR	Date
"Support for collection of more accurate real storage usage statistics" on page 64	PI78979	2017-08
"New LOAD utility IGNORE options " on page 64	PI77159	2017-08
"Real-time statistics collection of RUNSTATS-related columns starts at object creation" on page 64	PI79234	2017-07
"Explicitly-defined hidden ROWID columns" on page 65	PI77310	2017-06
"Db2 Native Rest Services " on page 65	PI70652	2017-05
"Db2 12 function level 501 activation" on page 65 • "The LISTAGG built-in function" on page 101	PI70535	2017-05
"ADMIN_INFO_SQL stored procedure enhanced to collect UDF information " on page 65	PI73268	2017-04
"LOAD utility support for more date and time formats" on page 65	PI69064	2016-11

New-function APARs for Db2 12 in 2024

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 12 in 2023

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 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(ALLCOMIT) 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 KEEPDYNAMIC 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 affect. 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 index traverse count information in DISPLAY STATS output

APAR PH55255 (August 2023) improves how Db2 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. This APAR retrofits new function that APAR PH51565 introduced in Db2 13 to Db2 12.

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.

For more information, see the following related topics:

- -DISPLAY STATS (Db2) (Db2 Commands)
- DSNT830I (Db2 Messages)
- Fast index traversal (Db2 Performance)
- PH55255

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:

Before this APAR, the DSNV491I message already included information such as 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.

For more information, see the following related topics:

- DSNV523I (Db2 Messages)
- DSNV491I (Db2 Messages)
- PH41756

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 DSN7GCMD
*** 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.

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
• -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.

For more information, see the following related topics:

• The REXX SQLDA (Db2 SQL)
• DESCRIBE OUTPUT (Db2 SQL)
• PH52852

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.

For more information, see the following related topics:

- PKG APPLCOMPAT field (Db2 Installation and Migration)
- PKG PLANMGMT field (Db2 Installation and Migration)
- PH50704

Improved default behavior for management of real storage

APAR PH47163 (February 2023) improves how Db2 12 manages real storage when the default setting AUTO is used for the REALSTORAGE_MANAGEMENT subsystem parameter. With the new behavior,

Db2 checks the available real storage on the LPAR, and uses the following threshold calculation to determine the management behavior to use:

```
(REALAVAIL > 1.2 * MAXSPACE)
```

REALAVAIL is the amount of currently available real storage on the LPAR, and MAXSPACE is a setting that restricts the virtual storage available to the z/OS DUMPSRV address space.

If the available storage is greater this threshold, Db2 now uses the behavior of the OFF setting and does not discard unused real storage at thread deallocation or during commit. However, if the amount of available real storage is lower than this threshold, Db2 switches to using the previous behavior (see the new AUTO1 setting description).

Db2 issues message DSNS006I when it switches behaviors.

The new AUTO behavior is preferable in most cases. However, if you encounter a situation that requires returning to the previous behavior, you can specify the new AUTO1 setting.

For more information, see the following related topics:

- MANAGE REAL STORAGE field (REALSTORAGE_MANAGEMENT subsystem parameter) (Db2 Installation and Migration)
- DSNS006I (Db2 Messages)

- PH47163

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.

For more information, see the following related topics:

- "EXPLAIN table changes" on page 188
- Migration step 25: Convert EXPLAIN tables to the current format (Db2 Installation and Migration)
- Activating Db2 12 new function at migration (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.

For more information, see the following related topics:

- DDF_COMPATIBILITY in macro DSN6FAC (Db2 Installation and Migration)
- PH50729

New-function APARs for Db2 12 in 2022

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.

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

Authorization update for CREATE INDEX on DGTT
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.

For more information, see the following related topics:

- CREATE INDEX (Db2 SQL)
- Creating declared temporary tables (Db2 Application programming and SQL)
- PH48601

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).

For more information, see the following related topics:

- PH48480

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 thatz/OS Unicode Conversion Services is used with only "normal" casing capabilities and use Unicode Standard 6.0.0.

For more information, see the following related topics:

- LOWER (Db2 SQL)
- UPPER (Db2 SQL)
- TRANSLATE (Db2 SQL)
- PH47187

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.

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 KEEPDYNAMIC(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 KEEPDYNAMIC-refreshable thread, and not a candidate for idle thread detection and termination. For more information, see Enabling KEEPDYNAMIC 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).

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.

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 REORG option to minimize the duration of the last log iteration

APAR PH33864 (May 2022) introduces a new option, LASTLOG,for both the REORG INDEX utility and the REORG TABLESPACE utility. You can use this new option to help reduce the outage window during REORG SHRLEVEL CHANGE executions. Specifically, LASTLOG controls whether the utility applies log records during the last log iteration in the LOG phase. By specifying LASTLOG NO, you can avoid costly subprocesses that occur during the final log iteration and potentially reduce the outage window. LASTLOG YES ensures that REORG completes the final round of applying log records. The default is LASTLOG YES, which is the existing behavior prior to this APAR.

For more information, see the following related topics:

- Syntax and options of the REORG INDEX control statement (Db2 Utilities)
- Syntax and options of the REORG TABLESPACE control statement (Db2 Utilities)
- PH33864

New LOAD option to reclaim NPI space

APAR PH39194 (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.

For more information, see the following related topics:

- Syntax and options of the LOAD control statement (Db2 Utilities)
- PH39194

Improvements to the -START ML and -STOP ML commands

With APAR PH43479 (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.

For more information, see the following related topics:

- -START ML (Db2) (Db2 Commands)
- -STOP ML (Db2) (Db2 Commands)
- PH43479

New statistics trace classes for monitoring distributed user statistics

With APAR PH40244 (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.

APAR PH40243 (April 2022), enhances the Distributed connection control (DCC) function of IBM Db2 AI for z/OS. You can enable IFCID 411 or 412 to collect thread-related statistics for client user IDs or client application names and use the statistics as input for DCC training.

For more information, see the following related topics:

- Preparing to monitor distributed connections (Db2ZAI)
-
- PH40243

Improved detection and removal of incomplete connections to Db2

APAR PH44270 (April 2022) improves how Db2 detects and terminates incomplete connections, and it adds new a message DSNL079I that Db2 issues to indicate that it terminated incomplete connections, when the required network exchanges between Db2 and the client do not process in a timely manner. Db2 issues the DSNL079I message at most once every 5 minutes. The DSNL079I indicates the number of connections terminated since it was last issued.

This APAR also enhances the DISPLAY LOCATION command with a keyword INCOMPLT. When INCOMPLT is specified, the output messages include a list of locations that had incomplete connections terminated.

For more information, see the following related topics:

- -DISPLAY LOCATION (Db2) (Db2 Commands)
- DSNL079I (Db2 Messages)
- DSNL200I (Db2 Messages)
- PH44270

DCLGEN updated to use COMP-5 declarations for integer data for COBOL applications

With APAR PH42857 (March 2022), the Db2 declarations generator (DCLGEN) now uses COMP-5 for declarations for integer data types, including SMALLINT, INTEGER, and BIGINT. This change simplifies the generation of declarations for COBOL programs when the TRUNC(OPT) option is used in an Enterprise COBOL Version 5 or later compiler. TRUNC(OPT) provides significant performance advantages over the other TRUNC options.

Before this change, if you wanted to benefit from the performance advantages of the TRUNC(OPT) compiler option and use DCLGEN, you needed to use one of the following manual workarounds, by modifying the DCLGEN copybooks:

- Change COMP to COMP-5 so it is handled as native binary.
- Add a digit to the definition, such as by changing PIC S9(4) to PIC S9(5), so that the data fits into the underlying picture clause.

Otherwise, you were forced to continue using the TRUNC(BIN) compiler option, and thus not benefit from the performance improvements available with TRUNC(OPT).

For more information, see the following related topics:

- DCLGEN (declarations generator) (Db2 Application programming and SQL)
- Data types that DCLGEN uses for variable declarations (Db2 Application programming and SQL)
- TRUNC compiler option (Enterprise COBOL for z/OS)

- PH42857

Statistics class 1 includes IFCID 0369

With APAR PH43916 (March 2022), IFCID 0369 is added to statistics class 1, and is activated by default. A trace for IFCID 0369 records wait time and CPU time for threads that run in a Db2 subsystem, aggregated by connection type. In previous Db2 releases, IFCID 0369 was in statistics class 9, and was not activated by default.

For more information, see the following related topics:

- -START TRACE (Db2) (Db2 Commands)
- PH43916

Controlling application compatibility for productivity-aid sample programs

APAR PH41968 (February 2022) introduces a new PKGSET parameter for the DSNTEP2, DSNTEP4, DSNTIAD, and DSNTIAUL sample programs. When you specify this parameter, the sample programs implicitly issue SET CURRENT PACKAGESET statements before processing the dynamic SQL statements.

With this capability, you can bind packages in specific collections for different application compatibility (APPLCOMPAT) levels, and use the PKGSET parameter to specify the package set, and thus the APPLCOMPAT level, that the program uses when it processes dynamic SQL statements.

In the DSNTEP2 and DSNTEP4 programs, you can also change the APPLCOMPAT level for different statements within a single SYSIN by specifying a `--#PKGSET package-set` control statement between the statements.

For more information, see the following related topics:

- Application compatibility levels in Db2 12 (Db2 Application programming and SQL)
- DSNTEP2 and DSNTEP4 sample programs (Db2 Application programming and SQL)
- DSNTIAD sample program (Db2 Application programming and SQL)
- DSNTIAUL sample program (Db2 Application programming and SQL)
- PH41968

LOAD improvements for inline image copies of partitions

With APAR PH40709 (January 2022), you can now request that the LOAD utility take partition-level sequential image copies by specifying the copy options only once, at the table space level. Previously, to get partition-level copies, you had to specify the copy options in each INTO TABLE PART clause.

This new syntax has the benefit of serialization at the partition level while still applying the global copy options to all partition-level copies. Serializing at the partition level, instead of the table space level, allows concurrent LOAD jobs to run against other partitions within the same table space. Applying global copy options to each partition copy can also help with resource constraints. For example, instead of needing 200 tape drives or data sets to take inline copies of 200 partitions, LOAD can use the single tape drive or data set that is specified in the copy options at the table space level. Prior to this APAR, you could not get both of these benefits in the same LOAD job. To get partition-level serialization, you had to specify the copy options in the INTO TABLE PART clauses. To apply global copy options, you had to specify them at the table space level. You could not do both.

The existing syntax is still valid. The new syntax introduced by this APAR is another option for requesting inline copies of partitions.

- Syntax and options of the LOAD control statement (Db2 Utilities)
- PH40709

New-function APARs for Db2 12 in 2021

Consistent inline image copies with LOAD REPLACE

With APAR PH39300 (December 2021) The LOAD utility is enhanced so that if you specify REPLACE or RESUME NO, any requested inline sequential image copies are consistent and usable as a recovery point. Sequential image copies are requested by the existing LOAD options COPYDDN and RECOVERYDDN.

Previously, for LOAD to create consistent sequential image copies, you had to specify RESUME YES with SHRLEVEL NONE. (SHRLEVEL CHANGE is not valid with COPYDDN and RECOVERYDDN.) If you specified REPLACE or RESUME NO, any sequential image copies did not include changed pages caused by updates during the INDEXVAL and ENFORCE utility phases. With this APAR, any sequential image copies that are taken during a LOAD REPLACE or RESUME NO operation now include updates during the INDEXVAL and ENFORCE phases. Therefore, these copies are consistent and valid as recovery points.

This enhancement is automatic and does not require any new keywords or changes to the LOAD statement syntax.

For more information, see the following related topics:

- Taking an inline COPY with LOAD (Db2 Utilities)
- Syntax and options of the LOAD control statement (Db2 Utilities)
- PH39300

Acceleration support for the MUTLIPLY_ALT built-in function

APAR PH42524 introduces acceleration support for the MULTIPLY_ALT built-in function with IBM Db2 Analytics Accelerator 7.1 or later.

For more information, see the following related topics:

- MULTIPLY_ALT (Db2 SQL)
- Supported SQL functions and expressions
- PH42524

Reduced console messages for FTB

APAR PH41751 (December 2021) eliminates message DSNI070I to reduce the number of console messages issued when fast index traversal (FTB) is used in Db2 12. The removed DSNI070I message reported changes to the number of objects using FTB and other FTB-related status information for a Db2 subsystem. Customers reported seeing too many of these messages.

To obtain detailed information about FTB use in Db2, you can use the IFCID 389 and IFCID 477 trace information, or issue the DISPLAY STATS command with the INDEXMEMORYUSAGE option.

For more information, see the following related topics:

- Fast index traversal (Db2 Performance)
- -DISPLAY STATS (Db2) (Db2 Commands)
- PH41751

Support for disabling the filtering of alter or modify log record collection in IFCID 306 trace applications

APAR PH40937 (December 2021) introduces the capability for applications that collect IFCID 306 trace records to receive alter or modify log records for all table spaces. Filtering by DBID and PSID for those log records is not performed. This capability is primarily for support of IBM Db2 Analytics Accelerator V7.5 using IBM Integrated Synchronization, but any application that collects IFCID 306 records can use it.

For more information, see the following related topics:

- Qualification fields for READS requests (Db2 Performance)
- Qualifying log records for IFCID 0306 requests (Db2 Administration Guide)

- PH40937

Longer password phrase support for connections to Db2 for z/OS

APAR PH40443 (November 2021) enhances the CONNECT statement to support longer password phrases for applications that connect to Db2 for z/OS. The *authorization* USING *host-variable* clause of the CONNECT statement now accepts password phrases up to 100 characters. With this enhancement, password phrases are also now supported for applications that use IBM Data Server Driver for JDBC and SQLJ type 2 connectivity for connections to Db2 for z/OS. However, the previous 8-character limit still applies for establishing trusted connections by using type 2 connectivity.

For more information, see the following related topics:

- CONNECT (Db2 SQL)
- Encrypted password, user ID, or data security under the IBM Data Server Driver for JDBC and SQLJ (Db2 Application Programming for Java)
- PASSWORD or PHRASE command (z/OS)
- PH40443

Improved identification of running log reader tasks from Integrated Synchronization products

APAR PH40261 (October 2021) enhances log reader related messages to identify the location and correlation ID of the IBM Db2 Analytics Accelerator for z/OS or the IBM Db2 for z/OS Data Gate instance that started the log reader task.

The APAR also adds a **SCOPE(GROUP)** option to the **-DISPLAY STATS (LOGREADERTASKS)** command to display information about log reader tasks running on all active members of a data sharing group.

For more information, see the following related topics:

- -DISPLAY STATS (Db2) (Db2 Commands)
- DSNT788I (Db2 Messages)
- DSNI090I (Db2 Messages)
- DSNI091I (Db2 Messages)
- DSNI092I (Db2 Messages)
- PH40261

The SYSPROC.ADMIN_COMMAND.DSN stored procedure now supports DCLGEN

With APAR PH28863 (August 2021) DCLGEN generates a table or view declaration and puts it into a member of a partitioned data set that you can include in your program.

Prior to PH28863, the DCLGEN subcommand could be issued only from a DSN session running in either foreground or background mode, or through DB2I. With PH28863, you can also use SYSPROC.ADMIN_COMMAND_DSN to execute the DCLGEN subcommand.

This enhancement provides modern IDEs with a simple method for generating language structures for data objects in the source code. Supporting DCLGEN from a stored procedure also enables REST services to be built around it. This new capability also reduces the z/OS skills that are needed to create and submit a JCL job that executes DCLGEN.

For more information, see the following related topics:

- ADMIN_COMMAND_DSN stored procedure (Db2 SQL)
- DCLGEN (DECLARATIONS GENERATOR) (DSN) (Db2 Commands)
- PH28863

Support for new SUBSTR_COMPATIBILITY subsystem parameter

APAR PH36071 (August 2021) introduces support for the SUBSTR_COMPATIBILITY subsystem parameter, which specifies whether the Db2 built-in function SUBSTR always returns an error message for invalid input.

Previously, during execution of the SUBSTR built-in function, Db2 sometimes incorrectly returned a result for invalid input instead of issuing an appropriate error message. After the PTF for APAR PH36071 is applied and Db2 12 function level 500 or higher is activated, the SUBSTR_COMPATIBILITY subsystem parameter is set to PREVIOUS by default and Db2 continues to behave as before the PTF was applied. If the SUBSTR_COMPATIBILITY subsystem parameter is set to CURRENT, Db2 always enforces the rules for the SUBSTR built-in function that are documented in SUBSTR (Db2 SQL) and returns an SQL error code if the rules are not met.

Before you set the SUBSTR_COMPATIBILITY subsystem parameter to CURRENT, you might need to modify some of your applications to handle this change.

For more information, see the following related topics:

- SUBSTR COMPATIBILITY field (SUBSTR_COMPATIBILITY subsystem parameter) (Db2 Installation and Migration)
- "Application and SQL release incompatibilities" on page 123
- DSNTIPX: Routine parameters panel (Db2 Installation and Migration)
- PH36071

IBM Text Search for Db2 for z/OS V5.5.4

IBM Text Search for Db2 for z/OS V5.5.4 is now available, after you apply the PTF for APAR PH35720 (August 2021) in Db2 12.

For more information, see the following related topics:

- IBM Text Search for Db2 for z/OS (IBM Text Search for Db2 for z/OS Installation, Administration, and Reference)
- Installing and configuring text search functions (IBM Text Search for Db2 for z/OS Installation, Administration, and Reference)
- IBM Text Search for Db2 for z/OS downloads
- PH35720

Improved APREUSE bind option functionality

APAR PH36728 (June 2021) provides new functionality to improve the reusability of certain access paths with the APREUSE bind option when query transformations are involved.

This reduces the likelihood that the APREUSE bind option will result in a bind error, or a bind warning.

For more information, see the following related topics:

- APREUSE bind option (Db2 Commands)
- Analyzing access path changes at bind or rebind (Db2 Performance)
- PH36728

IBM Db2 for z/OS Developer Extension 1.2

Announcing IBM Db2 for z/OS Developer Extension which delivers a new developer experience for Db2 for z/OS application development in a free, lightweight, easy-to-install, and easy-to-use package, integrated with today's most widely used IDE, Microsoft Visual Studio Code.

Support for Redirected Recovery with the RECOVER utility

APAR PH35266 (June 2021) enhances the RECOVER utility to support redirected recovery of index spaces and indexes using the recovery resources of different index spaces or indexes.

With redirected recovery, Db2 system administrators can more accurately determine how long the recoveries will take, check for recovery issues, and generate a replica of their production data at different points in time with transactional consistency without affecting the production data and without impact to production applications. Redirected recovery can also be used to evaluate whether recovering indexes from image copies plus application of log records is faster than rebuilding the indexes.

The FROM option of the RECOVER utility can now be used to restore image copies and apply Db2 log records from source index spaces or indexes to target index spaces and indexes. Both the source and target indexes and index spaces must have the COPY YES attribute.

For more information, see the following related topics:

- RECOVER (Db2 Utilities)

Support for limiting image copy data sets for REORG

With APAR PI75518 (June 2021) the REORG utility can combine data from multiple partitions into a single image copy data set, and you can now specify limits for the maximum number of image copy data sets the utility can allocate. To specify the limits, you can use two new keywords in the REORG utility control statement:

ICLIMIT_DASD specifies the maximum number of sequential image copies that REORG TABLESPACE can allocate to DASD, from 0 to 32767. 0 indicates no limit. ICLIMIT_DASD overrides the value of the REORG_IC_LIMIT_DASD subsystem parameter.

ICMLIMT_TAPE specifies the maximum number of sequential image copies that REORG TABLESPACE can allocate to tape, from 0 to 32767. 0 indicates no limit. ICLIMIT_TAPE overrides the value of the REORG_IC_LIMIT_TAPE subsystem parameter.

Before this APAR, REORG was able to create a single inline image copy for all target base partitions being reorganized, or create a separate image copy for each base partition. When reorganizing a large number of partitions or the entire partitioned table space, neither option provided the flexibility needed to operate under external constraints such as the number of available tape drives or memory consumption.

For more information, see the following related topics:

- Syntax and options of the REORG TABLESPACE control statement (Db2 Utilities)
- REORG IC LIMIT DASD field (REORG_IC_LIMIT_DASD subsystem parameter) (Db2 Installation and Migration)
- REORG IC LIMIT TAPE field (REORG_IC_LIMIT_TAPE subsystem parameter) (Db2 Installation and Migration)
- PI75518

SQL tuning capabilities added to Db2 Accessories Suite 4.2

IBM Db2 Accessories Suite for z/OS 4.2 has been enhanced with the Database Services Expansion Pack feature, which includes IBM SQL Tuning Services.

SQL Tuning Services provides a series of APIs that can be incorporated into your continuous integration/continuous delivery (CI/CD) process to help ensure that your SQL is optimally tuned before new code is deployed. SQL Tuning Services includes APIs for capturing queries, generating visual representations of access plans for a query, generating diagnostic information, and generating statistics advice, as well as a collection of administrative APIs for setting up and administering SQL Tuning Services.

For more information, see the following related topics:

- Overview of IBM SQL Tuning Services (IBM SQL Tuning Services)

Fast index traversal (FTB) support for non-unique indexes

With APAR PH30978 (June 2021) applied, Db2 12 at function level 508 or higher now supports fast index traversal (FTB) for non-unique indexes.

FTBs provide valuable Db2 transaction performance improvements without the need for application changes. FTBs use a cache-friendly optimized in-memory structure to accelerate index-tree traversals, which can significantly reduce get page operations and CPU consumption. They are enabled by default in Db2 12 by the INDEX_MEMORY_CONTROL subsystem parameter setting and can take effect as soon as function level 100.

To enable use of FTBs for non-unique indexes, you set the value of the FTB_NON_UNIQUE_INDEX subsystem parameter to YES. The default value is NO.

For more information, see the following related topics:

- Fast index traversal (Db2 Performance)
- FTB NON UNIQUE INDEX field (FTB_NON_UNIQUE_INDEX subsystem parameter) (Db2 Installation and Migration)
- "Function level 508 (PH29392 - October 2020)" on page 75
- PH30978

Improved REORG INDEX utility performance

To improve performance and resource consumption, with APAR PH25217 (May 2021), REORG INDEX utility jobs with SHRLEVEL REFERENCE or SHRLEVEL CHANGE no longer require a work data set to hold the unloaded index keys. Instead, the utility can store these keys in memory. REORG INDEX can also now use parallel subtasks to unload and build the index keys and partitions in parallel. To specify that you want REORG INDEX to use this new behavior, specify the new NOSYSUT1 option in your utility statement or set the new subsystem parameter REORG_INDEX_NOSYSUT1 to YES. The default is NO.

If you set REORG_INDEX_NOSYSUT1 to YES or add NOSYSUT1 to your REORG INDEX statements, you do not have to remove the WORKDDN option from existing utility statements or modify existing JCL. In this case, REORG INDEX ignores the WORKDDN option and the associated template or JCL DD statement (SYSUT1 by default).

REORG INDEX with SHRLEVEL NONE is not affected by this new behavior.

For more information, see the following related topics:

- Syntax and options of the REORG INDEX control statement (Db2 Utilities)
- REORG INDEX NOSYSUT1 (REORG_INDEX_NOSYSUT1 subsystem parameter) (Db2 Installation and Migration)
- PH25217

RESTSERVICEDEFAULT option for REBIND PACKAGE

APAR PH34544 (May 2021) introduces a new RESTSERVICEDEFAULT bind option, which can be used with the REBIND PACKAGE DSN subcommand to change a REST service "default service version" attribute without causing unnecessary application outages. The new RESTSERVICEDEFAULT bind option is only allowed for the REBIND PACKAGE DSN subcommand and only when the package being rebound is a REST service package.

Previously, Db2 REST services versioning support set the first created instance of each service as being the "default" service version and there was no way to change the default version of a REST Service.

For more information, see the following related topics:

- RESTSERVICEDEFAULT bind option (Db2 Commands)
- PH34544

Display information about use of fast index traversal (FTB) with DISPLAY STATS

With APAR PH34859 (May 2021), you can issue a DISPLAY STATS command and specify the new INDEXTRAVERSECOUNT option to display information about the use of fast index traversal (FTBs) for a specific index, or a list of the first *n* indexes with the most uses of FTBs. Db2 returns the results in message DSNT830I.

For more information, see the following related topics:

- Fast index traversal (Db2 Performance)
- -DISPLAY STATS (Db2) (Db2 Commands)

- DSNT830I (Db2 Messages)
- PH34859

Db2 12 function level 510 activation

Activating function level 510 (PH33727 - April, 2021) enforces the completion of various pre-migration activities before migration to Db2 13, and is required before you can migrate a Db2 12 environment to the Db2 13. It does not introduce any new features or capabilities, or catalog changes, in Db2 12.

For more information, see the following related topics:

- "Function level 510 (V12R1M510) (PH33727 - April 2021)" on page 69
- PH33727
- IBM Db2 13 for z/OS (announcement letter)

REORG utility improvements for SORTNPSI YES

Previously, when you specified the REORG utility option SORTNPSI YES or set the REORG_PART_SORT_NPSI subsystem parameter to YES, REORG sorted the non-partitioned secondary index (NPSI) keys only when certain internal thresholds were met. If the index keys are not sorted, REORG cannot collect inline statistics. Therefore, in those cases where REORG determined not to sort the NPSI keys, you had to run the RUNSTATS utility separately to collect statistics.

With PH34403 (April 2021), REORG sorts the keys for NPSIs when you specify the SORTNPSI YES utility option or set the REORG_PART_SORT_NPSI subsystem parameter to YES. Because the keys are always sorted, inline statistics can always be collected.

For more information, see the following related topics:

- Syntax and options of the REORG TABLESPACE control statement (Db2 Utilities)
- REORG PART SORT NPSI field (REORG_PART_SORT_NPSI subsystem parameter) (Db2 Installation and Migration)
- "Db2 utility release incompatibilities" on page 134
- PH34403

Increased password size for Db2-supplied stored procedures

APAR PH35411 (April 2021) increases the size of the input parameter PASSWORD for the following Db2-supplied stored procedures from VARCHAR(24) to VARCHAR(100), and enables these stored procedures to accept 100-character passphrases.

For more information, see the following related topics:

- ADMIN_COMMAND_UNIX stored procedure (Db2 SQL)
- ADMIN_JOB_CANCEL stored procedure (Db2 SQL)
- ADMIN_JOB_FETCH stored procedure (Db2 SQL)
- ADMIN_JOB_QUERY stored procedure (Db2 SQL)
- ADMIN_JOB_QUERY stored procedure (Db2 SQL)
- PH35411

Python support on Db2 for z/OS through ODBC driver with IBM_DB Pyhton support for DB2 for z/OS

APAR PH30575 (March 2021) enables Python applications to access data in Db2 for z/OS data by using the IBM_DB Pyhton support for DB2 for z/OS driver through Db2 for z/OS ODBC driver CLI APIs.

For more information, see the following related topics:

- Python support for Db2 for z/OS (Db2 Programming for ODBC)
- IBM_DB Python Support for IBM Db2 for z/OS and IBM Informix databases
- PH30575

Clean-up phased-out copies for rebind phase-in

APAR PH33295 (March 2021) provides new options to free unused phased-out copies which could be created when a package is rebound. The freeing of phased-out package copies is recommended in order to reduce space in the Db2 directory and catalog.

For more information, see the following related topics:

- Package copies for plan management (Db2 Performance)
- FREE PACKAGE (DSN) (Db2 Commands)
- PH33295

Real-time monitoring of CICS attachment facility calls

When z/OS Connect calls a REST API to invoke a CICS application that accesses Db2 through the CICS attachment facility, and Db2 APAR PH31447 (February 2021) and CICS APAR PH30252 are applied, CICS passes new identifying client information to Db2. That information can be recorded in Db2 trace records, which you can correlate with corresponding trace records from CICS to clearly identify the origin of requests from such clients.

The new CICS client information is also available through the Db2 CURRENT CLIENT_APPLNAME and CURRENT CLIENT_ACCTNG special registers.

For more information, see the following related topics:

- CURRENT CLIENT_APPLNAME special register (Db2 SQL)
- CURRENT CLIENT_ACCTNG special register (Db2 SQL)
- PH31447
- PH30252

Db2 12 function level 509 activation

Function level 509 (PH33015 - January, 2021) introduces support for tamper-proof audit policies, high availability for accelerator-only tables, the specification of a compression algorithm at the object level, and a temporal RI enhancement to allow UPDATE or DELETE on a parent table.

For more information, see the following related topics:

- "Function level 509 (PH33015 - February 2021)" on page 71
- PH33015

Improved transaction execution times and concurrency for REBIND PACKAGE

FL 505 APAR PH28693 (January 2021) improves concurrency for REBIND commands in Db2 12 at function level 505 or higher. With this APAR, a REBIND command now always obtains a U lock, allowing subsequent transactions that are executing a package to run in parallel. This change does not apply when the package is for a trigger, a native SQL stored procedure, or a UDF.

The change applies to packages rebound with the following options:

- APREUSE(NONE) PLANMGMT(EXTENDED)
- APREUSE(WARN) PLANMGMT(EXTENDED) APREUSESOURCE(CURRENT)
- APREUSE(ERROR) PLANMGMT(EXTENDED) APREUSESOURCE(CURRENT)

For more information, see the following related topics:

- Package copies for plan management (Db2 Performance)
- "Rebind phase-in for packages that are being used for execution" on page 85
- PH28693

Enhanced DEFAULTIF specifications for the LOAD utility

In LOAD utility statements, you can specify that a default value be loaded into a column by specifying the DEFAULTIF clause. Prior to APAR PH30610 (January 2021), the only condition that you could specify in that clause is an "equal to" (=) comparison. For example:

```
LOAD … COL2 POSITIONS(5:10) CHAR(6) DEFAULTIF(COL1='ABC')
```

In this case, if COL1 equals ABC, the default column value is loaded into COL2.

With this APAR, the DEFAULTIF clause is enhanced, so that you can now specify the following additional conditions:

A default column value is loaded if a conversion error occurs. For example:

```
COL2 POSITION(5:14)DATE DEFAULTIF(CONV_ERROR)
```

The new CONV_ERROR keyword indicates that if a conversion error occurs when loading the column value, the default column value is to be used instead.

A default column value is loaded if a field does not equal (<>) a particular value. For example:

```
COL2 POSITION(5:10) CHAR(6) DEFAULTIF(COL1<>'ABC')
```

The new <> specification indicates that if COL1 does not equal ABC, the default value is loaded into COL2.

Similarly, the following example specifies that if the value from position 5 to position 10 does not equal ABCDE, the default value is loaded into COL2:

```
COL2 POSITION(5:10) CHAR(6) DEFAULTIF((5:9) <> 'ABCDE')
```

For more information, see the following related topics:

- Syntax and options of the LOAD control statement (Db2 Utilities)
- PH30610

LOAD REPLACE or REBUILD INDEX utilities now reset RTS for GETPAGES

With APAR PH32224 (January 2021), when the LOAD REPLACE or REBUILD INDEX utilities are run for an object, Db2 now resets the real-time statistics (RTS) value for GETPAGES in the SYSIBM.SYSTABLESPACESTATS and SYSIBM.SYSINDEXSPACESTATS catalog tables. This change makes the LOAD REPLACE and REBUILD index utility behavior consistent with the existing REORG utility behavior for the RTS GETPAGES values.

For more information, see the following related topics:

- How LOAD affects real-time statistics (Db2 Performance)
- How REBUILD INDEX affects real-time statistics (Db2 Performance)
- SYSINDEXSPACESTATS catalog table (Db2 SQL)
- SYSTABLESPACESTATS catalog table (Db2 SQL)
- PH30780

New-function APARs for Db2 12 in 2020

The following enhancements are introduced in the Db2 12 - initial release. That is, they are not controlled by the activation of function level 501 or higher. However, some of these enhancements might be controlled by the activation of function level 500, which is comparable to new-function mode in previous Db2 releases. They are listed roughly in order of availability, beginning with the most recent.

Integrated Accelerator for Z Sort use for GROUP BY or ORDER BY

The IBM z15® processor includes the Integrated Accelerator for Z Sort coprocessor, which can help to reduce CPU usage and improve elapsed time for eligible sort workloads.

With PH31684, when Db2 for z/OS applications run on a system that has the IBM Integrated Accelerator for Z Sort coprocessor, sort operations in queries that include GROUP BY or ORDER BY clauses can experience better performance.

For more information, see the following related topics:

- PH31684

Controlling all distributed threads and connections with profile tables

This APAR (PH30780 - December, 2020)introduces new profile table actions, MONITOR ALL CONNECTIONS and MONITOR ALL THREADS, in the KEYWORDS column of the DSN_PROFILE_ATTRIBUTES table. You can use the new keywords to specify warning and exception thresholds that apply _cumulatively_ across all connections and threads under the control of the default location profile. The new MONITOR ALL actions are independent of the existing MONITOR CONNECTIONS and MONITOR THREADS, which means they can be used together or separately.

Before this new capability, profile monitoring allowed limiting the number of connections and threads, through the use of existing keywords MONITOR CONNECTIONS and MONITOR THREADS, including for unknown applications on an individual IP address basis. However, it was not possible to limit the cumulative number of connections and threads for all unknown locations. An excessive number of unknown locations accessing Db2 can exceed system limits for connections or threads and cause an outage.

The new keywords can be specified only for profiles that use the default location filtering criteria. That is, the LOCATION column in DSN_PROFILE_TABLE contains one of the following values representing any IP address: '*', '::0', or '0.0.0.0'.

For more information, see the following related topics:

- Monitoring threads by using profile tables (Db2 Administration Guide)
- Monitoring remote connections by using profile tables (Db2 Administration Guide)
- DSN_PROFILE_ATTRIBUTES profile table (Db2 Performance)
- PH30780

TEMPLATE utility support for LBI

With PH30093 (December, 2020) TEMPLATE utility control statement is enhanced to support *large block interface* (LBI) when allocating data sets. To request LBI, specify the new BLKSZLIM keyword with an upper limit for the data set block size. When the data set is dynamically allocated,z/OS determines the optimal block size, not to exceed the specified upper limit.

Using LBI can help decrease CPU use and elapsed time for utilities and thus improve overall utility performance. LBI allows you to increase the block size for data sets and allows utilities to handle much larger blocks with BSAM. LBI is mainly useful when creating new data sets on tape.

For more information, see the following related topics:

- Syntax and options of the TEMPLATE control statement (Db2 Utilities)
- PH30093

Improved status messages for DDF

With APAR PH30222 (November 2020) applied, Db2 displays new and improved messages to indicate when you are approaching the MAXDBAT threshold and when you have exceeded it in real time:

- New message DSNL077I message displays when the current number of active, in-use DBATs exceed 85% of the current MAXDBAT value.
- New message DSNL078I message displays when the current number of active, in-use DBATs no longer exceeds 75% of the current MAXDBAT value.

- Existing message DSNL093I, which displays the result of the -DISPLAY DDF DETAIL command now includes the number of active in-use DBATs (IUDBAT), which is a calculation of the number of disconnected pooled DBATS (DSCDBAT) subtracted from the current number of active DBATS (ADBAT).
- Existing messages DSNL519I and DSNL523I can now be issued to display the availability of the DDF resync port, and the format of those messages is updated to indicate the kind of TCPIP port availability being displayed, including: PORT, SECURE PORT, or RESYNC PORT.

For more information, see the following related topics:

- DSNL077I (Db2 Messages)
- DSNL078I (Db2 Messages)
- DSNL093I (Db2 Messages)
- DSNL519I (Db2 Messages)
- -DISPLAY DDF (Db2) (Db2 Commands)
- PH30222

Authorized programs can pass eight-byte authorization IDs to Db2

Prior to APAR PH28017 (October 2020), when a connection to Db2 was requested, the maximum length of the authorization ID needed to be seven or fewer bytes. If the authorization ID was eight bytes, the last byte was truncated, which caused a connection request to Db2 to fail. This APAR allows IBM products, vendor tools, and other applications that are authorized programs to pass eight-byte authorization IDs to Db2.

For more information, see the following related topics:

- PH28017

Support for Redirected Recovery with the RECOVER utility

APAR PH27043 (October 2020) enhances the RECOVER utility to support redirected recovery of table spaces using the recovery resources of different table spaces. With redirected recovery, Db2 system administrators can more accurately determine how long the recoveries will take, check for recovery issues, and generate a replica of their production data at different points in time with transactional consistency without affecting the production data and without impact to production applications.

The new FROM option is added to the RECOVER utility to restore image copies and apply Db2 log records from source table spaces to target table spaces.

For more information, see the following related topics:

- RECOVER (Db2 Utilities)
- PH27043

Obsolete subsystem parameters are removed

APAR PH28280 (October 2020) is the last of a series of four that adjust Db2 subsystem parameters defaults to better align with current best practices, or remove obsolete subsystem parameters to simplify installation, migration, and configuration of Db2 subsystems. (The previous APARs are PH26317, PH24358, and PH21370.) APAR PH28280 removes the following Db2 subsystem parameters.

Removed parameter	Previous default	Resulting behavior
CACHEPAC	5242880	Db2 12 always allocates 10,485,760 bytes of storage for caching package authorization information for all packages on each Db2member. (However, if COMCRIT=YES, the amount of storage is set to 0 bytes.)

Removed parameter	Previous default	Resulting behavior
CACHERAC	5242880	Db2 12 always allocates 10,485,760 bytes of storage for caching routine authorization information for all routines on each Db2 member. (However, if COMCRIT=YES, the amount of storage is set to 0 bytes.)
IRLMAUT	YES	Db2 12 always automatically starts and stops the IRLM.
IRLMSWT	120	Db2 always waits a maximum of two minutes for IRLM to start automatically.
PCLOSEN	10	The interval for converting read-write page sets to read-only is always based on the PCLOSET setting.
MGEXTSZ	YES	Db2 12 always uses a sliding scale for sizing secondary extent allocations for Db2-managed data sets.
CHGDC	NO	The related data capture product is not used.
EDPROP	NO	The related data capture product is not used.

For more information, see the following related topics:

- Subsystem parameter (ZPARM) simplification in <keyword conref="../cmn/db2z_cmn_strings.dita#strings/newr"/>
- PH28280

Db2 12 function level 508 activation

Function level 508 (PH29392 - October 2020) introduces support for moving tables from deprecated multi-table simple and multi-table segmented table spaces to partition-by-growth universal table spaces (UTS), without taking an outage, to improve application and database operations and performance.

For more information, see the following related topics:

- "Function level 508 (PH29392 - October 2020)" on page 75
- PH29392

INSERT, UPDATE or DELETE statements prepared with the isolation-clause attribute WITH UR are disallowed

With APAR PH14791 (October 2020) applied, Db2 disallows INSERT, UPDATE or DELETE statements to be prepared with the isolation-clause attribute WITH UR. Db2 now issues SQLCODE -199 if with the WITH UR attribute is used to prepare any of the following statements:

- UPDATE WHERE CURRENT OF *cursor-name*
- DELETE WHERE CURRENT OF *cursor-name*
- INSERT without SELECT

Before this APAR, Db2 inadvertently allows the illegal syntax.

This APAR also introduces a new subsystem parameter ALLOW_UPD_DEL_INS_WITH_UR, which specifies whether Db2 disallows the illegal syntax. It accepts the following values:

NO - Disallows UPDATE, DELETE, or INSERT statements to be prepared with the isolation-clause attribute WITH UR. NO is the default value.

YES - Allow UPDATE, DELETE, or INSERT statements to be prepared with the isolation-clause attribute WITH UR.

For more information, see the following related topics:

- ALLOW_UPD_DEL_INS_WITH_UR in macro DSN6SPRM (Db2 Installation and Migration)
- PH14791

REORG TABLESPACE utility support for IBM Integrated Accelerator for Z Sort feature

Starting in Db2 12 function level 100, APAR PH28183 (September 2020) introduces the support for the Db2 REORG TABLESPACE utility to invoke DFSORT with the new IBM Integrated Accelerator for Z Sort feature available in the latest IBM z15 mainframe hardware and above. The new capability applies only to the variable length data sort for data records of a target table for the REORG TABLESPACE utility. It improves the CPU and elapsed time in the sort processing by DFSORT utilizing more in-memory sorting.

The new UTILS_USE_ZSORT subsystem parameter specifies whether Db2 utilities use the IBM Integrated Accelerator for Z Sort interface when invoking DFSORT for sort processing.

The UTILS_USE_ZSORT subsystem parameter supports the following values:

NO - Db2 utilities do not use integrated sort acceleration when invoking DFSORT.

YES - Db2 utilities attempt to use the integrated sort acceleration if possible when invoking DFSORT.

For more information, see the following related topics:

- UTILS USE ZSORT (UTILS_USE_ZSORT subsystem parameter) (Db2 Installation and Migration)
- PH28183
- IBM z15 and IBM z14® Frequently asked Questions

REPAIR DBD TEST and DIAGNOSE allowed on the work file database

With PH10358 (August 2020) The REPAIR utility is enhanced so that you can use the DBD statement with the TEST or DIAGNOSE options on the work file database DSNDB07. Previously, those options were restricted from being used with DSNDB07. REPAIR DBD REBUILD is still not allowed with DSNDB07.

For more information, see the following related topics:

- Repairing DBDs (Db2 Utilities)
- Syntax and options of the REPAIR control statement (Db2 Utilities)
- Work file database (Introduction to Db2 for z/OS)
- PH10358

Control of LOAD utility updates to MAXASSIGNEDVAL catalog values for identity columns

Starting in Db2 12 function level 100 with APAR PH28476 (August 2020), the LOAD utility always maintains the value of MAXASSIGNEDVAL in SYSIBM.SYSEQUENCES for any loaded identity columns. For some clients migrating to Db2 12, the new Db2 12 behavior is an incompatible change because of logic in their applications that relies on the previous behavior.

Prior to Db2 12, the LOAD utility did not maintain the value of MAXASSIGNEDVAL when the identity column value was provided by the user. The identity column value can be provided by the user if the column is defined as GENERATED BY DEFAULT or if the column is defined as GENERATED ALWAYS and the OVERRIDE(IDENTITY) option is specified in the LOAD statement.

With this APAR, you can disable the new behavior in Db2 12 by specifying UPDMAXASSIGNEDVAL NO in your LOAD statements. To aid your migration to Db2 12, you can begin adding this new option to your LOAD jobs in Db2 11. Although UPDMAXASSIGNEDVAL NO is accepted in Db2 11, it has no effect until Db2 12.

For more information, see the following related topics:

- "Db2 utility release incompatibilities" on page 134
- Syntax and options of the LOAD control statement (Db2 Utilities)
- PH28476

LOAD utility can load data into read-only objects

With APAR PH26131 (August 2020) You can now use the LOAD utility to load data into read-only objects if the LOAD_RO_OBJECTS subsystem parameter is set to Y. With this setting, LOAD is supported for read-only target objects, including base table, LOB, and XML table spaces and indexes. The default value for LOAD_RO_OBJECTS is N, which disallows loading data into read-only target objects with the LOAD utility.

Before APAR PH26131, Db2 always issued message DSNU590I with reason code 00C90080 for any attempt to load a read-only target object.

For more information, see the following related topics:

- LOAD RO OBJECTS field (LOAD_RO_OBJECTS subsystem parameter) (Db2 Installation and Migration)
- PH26131

MFA authentication support for clients not configured for sysplex WLB

APAR PH21341 (August 2020) introduces the capability to enable Db2 to store multi-factor authentication (MFA) credentials in the Db2 global authentication cache for distributed clients that do not use sysplex workload balancing (WLB) or seamless failover, and to specify the amount of time that the entry remains if unused.

The new MFA_AUTHCACHE_UNUSED_TIME subsystem parameter specifies how long in seconds that multi-factor authentication (MFA) security credentials from a distributed client (DRDA or REST) can remain unused in the Db2 global authentication cache before new security credentials must be provided.

An authentication cache entry is considered unused until its related MFA-based security credentials are reused by the same client IP address in a subsequent authentication request. The unused time is reset each time that the credentials are used. The authorization cache entry does not contain the text of the security credentials.

For more information, see the following related topics:

- Enabling caching of MFA and RACF PassTickets credentials for clients without sysplex workload balancing (Managing Security)
- MFA AUTH UNUSED TIME field (MFA_AUTHCACHE_UNUSED_TIME subsystem parameter) (Db2 Installation and Migration)
- DSN3583I (Db2 Messages)
- PH21341

New option to force the canceling of claimers during LOAD

To avoid situations where the LOAD utility fails because it could not gain exclusive control over the target object, APAR PH24369 (August 2020) introduces the new FORCE option for LOAD. This option is similar to the existing FORCE option for the REORG TABLESPACE utility. You can use it to force the cancellation of blocking claimers during drain processing to ensure that the utility completes.

For more information, see the following related topics:

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

- PH24369

CHECK utilities with SHRLEVEL CHANGE reset CHECK-pending status

Before APAR PH25593 (August 2020), if you specified the SHRLEVEL CHANGE utility option, the CHECK DATA and CHECK LOB utilities could not reset the following restrictive states:

- CHECK-pending (CHKP)
- Auxiliary CHECK-pending (ACHKP)
- Auxiliary warning (AUXW)

With PH25593, if you run CHECK DATA with SHRLEVEL CHANGE and no inconsistencies are found, any existing CHKP, ACHKP, or AUXW states on the objects are reset. Similarly, if you run CHECK LOB with SHRLEVEL CHANGE and no inconsistencies are found, any existing CHKP or AUXW states on the objects are reset. (Note that ACHKP applies to base table spaces only, and therefore cannot be reset by CHECK LOB.)

For more information, see the following related topics:

- Resetting CHECK-pending status (Db2 Utilities)
- PH25593

LOAD option to define auxiliary objects

With APAR PH19072 (August 2020), the LOAD utility can now allocate the associated data sets for target auxiliary objects with the DEFINE NO attribute, regardless of whether these objects are populated by LOAD. To specify that LOAD is to allocate data sets for DEFINE NO LOB and XML objects, specify the new utility option DEFINEAUX YES. The default behavior is the previous LOAD behavior: the utility allocates data sets for DEFINE NO auxiliary objects only when data is loaded into them.

For more information, see the following related topics:

- Syntax and options of the LOAD control statement (Db2 Utilities)
- PH19072

Availability of IBM Db2 for z/OS Developer Extension

The first iteration of IBM Db2 for z/OS Developer Extension for Visual Studio Code (Db2 Developer Extension) is available as a free download from the [Microsoft Visual Studio Code Marketplace] (https://marketplace.visualstudio.com/items?itemName=IBM.db2forzosdeveloperextension).

This extension provides application developers with language support for the SQL syntax that is used to define, manipulate, and control data in Db2 for z/OS databases. It is supported by Visual Studio Code and by the [Eclipse Theia](https://theia-ide.org/) IDE platform.

OBJECT_CREATE_FORMAT and UTILITY_OBJECT_CONVERSION subsystem parameters removed

APAR PH26317 (July 2020) removes the following Db2subsystem parameters to simplify installation, migration, and configuration of Db2subsystems.

Db2now always applies the following settings:

OBJECT_CREATE_FORMAT=EXTENDED
All new objects are created in the extended 10-byte RBA or LRSN format.

UTILITY_OBJECT_CONVERSION=NOBASIC
The LOAD, REBUILD INDEX, or REORG utilities always convert the target object in the 6-byte RBA or LRSN format to the extended 10-byte format.

Important: When the NOBASIC setting of the UTILITY_OBJECT_CONVERSION subsystem parameter is in effect, any LOAD, REORG, and REBUILD INDEX utility job that specifies RBALRSN_CONVERSION BASIC always fails. Similarly, if the target object is still in the 6-byte page format, these utility jobs also fail if they specify RBALRSN_CONVERSION NONE. Plan to remove the RBALRSN_CONVERSION option from these utility jobs before applying the PTF for PH26317.

For more information, see the following related topics:

- "The extended 10-byte RBA and LRSN in Db2 12" on page 209
- "Subsystem parameter changes in Db2 12" on page 200
- PH26317

Message DSNS005I now issued to help explain why storage contraction began

APAR PH25940 (July 2020) adds message DSNS005I to explain what situation caused Db2to begin storage contraction. DSNS005I is issued after DSNV516I, which indicates that STORAGE CONTRACTION MODE is beginning. Storage contraction is often related to the setting of the REALSTORAGE_MANAGEMENT subsystem parameter.

For more information, see the following related topics:

- DSNS005I (Db2 Messages)
- MANAGE REAL STORAGE field (REALSTORAGE_MANAGEMENT subsystem parameter) (Db2 Installation and Migration)
- PH25940

LOAD can override row change timestamp columns that are defined as GENERATED ALWAYS

You can now use the LOAD utility to load data into row change timestamp columns that are defined as GENERATED ALWAYS. For example, if you unload data from a table with a row change timestamp column, you can preserve those timestamp values when loading the data into the same or another table. Previously, if the row change timestamp column in the target table was defined as GENERATED ALWAYS, the utility could not load those timestamp values. With APAR PH25572 (July 2020), you can now override this GENERATED ALWAYS column by specifying the new OVERRIDE(ROWCHANGE) option in the LOAD statement.

For more information, see the following related topics:

- Syntax and options of the LOAD control statement (Db2 Utilities)
- PH25572

LOAD can presort data

With APAR PH23105 (June 2020), the LOAD utility can now sort input records in clustering order before loading a table space. Therefore, after the utility operation is complete, the loaded records are in clustering order without requiring you to take any additional action. (Existing records in the table space are not sorted.) To specify that you want LOAD to sort the input records first, use the new PRESORT option.

Having records in clustering order can improve the performance of certain SQL operations such as grouping, ordering, or comparisons other than equal. Previously, to ensure that loaded data was in clustering order, you had to either run the REORG utility after loading the data or add a prior step in your JCL to presort the input data. This APAR eliminates those extra steps.

For more information, see the following related topics:

- Syntax and options of the LOAD control statement (Db2 Utilities)
- PH23105

GETPAGES columns in RTS tables now wrap after reaching the largest supported value

APAR PH22838 (June 2020) changes the GETPAGES columns of the real-time statistics (RTS) tables, SYSIBM.SYSTABLESPACESTATS and SYSIBM.SYSINDEXSPACESTATS, so that the values now wrap after they reach the maximum supported positive BIGINT value, which is 9223372036854775807.

Before now, the GETPAGES values in the RTS tables remained static after reaching the maximum value, until the next REORG reset it to 0. However, the static maximum values were not useful for clients who wanted to monitor ongoing getpage activity in specific intervals. Now you can continue seeing getpage activity after the maximum values was reached. To determine when the wraparound occurred, you can analyze the temporal history tables for the RTS tables.

For more information, see the following related topics:

- SYSINDEXSPACESTATS catalog table (Db2 SQL)
- SYSTABLESPACESTATS catalog table (Db2 SQL)
- PH22838

IBM Db2 AI for z/OS 1.3.0

APARs PH15742, PH16111, PH24097, and PH25342 (June 2020) introduce support for Db2ZAI 1.3.0. Additional capabilities added by this version of Db2ZAI include automated SQL regression detection and resolution. Db2ZAI will now detect when an access path has degraded compared to a prior access path, and will automatically lock-in and fall back to the better-performing access path. This self-healing capability enables Db2ZAI to resolve regressions without intervention by a database administrator, thus saving valuable time and effort.

Another change with this version is that Db2ZAI 1.3.0 now provides an enhanced exception analysis experience, in which exceptions are categorized into general regions of interest to system programmers (such as CPU, storage, and connectivity) to allow a more immediate identification of problem areas.

Db2ZAI 1.3.0 also introduces distributed connection control, which provides protection for Db2 for z/OS from unintentional disruptions and intrusions that can originate from remote applications.

For more information, see the following related topics:

- IBM Db2 AI for z/OS 1.3.0 documentation
- PH15742
- PH16111
- PH24097
- PH25342

LOAD support for dates in 3-byte packed decimal format

With APAR PH22944 (June 2020), you can now use the LOAD utility to load data that is the 3-byte packed decimal representation of a date in DATE_P format (X'YYDDDs'). To specify this format, use the new DATE DATE_P option in the field specification. For details on this new syntax, see Syntax and options of the LOAD control statement (Db2 Utilities).

For more information, see the following related topics:

- PH22944

Db2 12 function level 507 activation

Function level 507 (PH24371 - June 2020) introduces application granularity for locking limits, deletion of old statistics when using profiles, CREATE OR REPLACE for procedures, and new passthrough-only and acceleration support with IBM Db2 Analytics Accelerator.

For more information, see the following related topics:

- "Function level 507 (PH24371 - June 2020)" on page 77
- PH24371

Enable fast index traversal (FTBs) for specific indexes only

APAR PH23238 (June 2020) introduces the capability to enable the fast index traversal capability (sometimes also called "fast traverse blocks" or "FTBs") for specific selected indexes. The general intent is for the use of FTBs to be controlled by Db2 through automation. However, clients have requested the ability to limit the use of FTBs to specific indexes, to enable focused testing as they prepare to exploit FTBs at scale.

To support that request, this APAR enhances the INDEX_MEMORY_CONTROL subsystem parameter with new settings. When you specify YES in the FOR SELECTED INDEXES ONLY field on panel DSNTIP71, Db2 uses the following settings:

- `INDEX_MEMORY_CONTROL=(SELECTED,AUTO)`, if you specify AUTO for the INDEX MEMORY CONTROL field.
- `INDEX_MEMORY_CONTROL=(SELECTED,n)`, where *n* is the storage limit specified in the INDEX MEMORY CONTROL field, in the range 10 - 200000 MB.

When either type of SELECTED setting is used, you control the specific indexes that can use FTBs by inserting a rows for the indexes with ACTION='A' in the SYSIBM.SYSINDEXCONTROL catalog table.

For more information, see the following related topics:

- Enabling or disabling fast index traversal at the index level (Db2 Performance)
- DSNTIP71: SQL OBJECT DEFAULTS PANEL 2 (Db2 Installation and Migration)
- INDEX MEMORY CONTROL field (INDEX_MEMORY_CONTROL subsystem parameter) (Db2 Installation and Migration)
- SYSINDEXCONTROL catalog table (Db2 SQL)
- PH23238

COMPRESS_SPT01 and SPT01_INLINE_LENGTH subsystem parameters removed

APAR PH24358 (June 2020) removes the following Db2 subsystem parameters to simplify installation, migration, and configuration of Db2 subsystems.

Db2 now always applies the following settings:

- **COMPRESS_SPT01=YES**: Db2 compresses the SPT01 table space during the next reorganization of the table space.
- **SPT01_INLINE_LENGTH=32138**: this setting is the maximum size of LOB column data that is maintained in base tables in the SPT01 directory table space.

For more information, see the following related topics:

- "Subsystem parameter changes in Db2 12" on page 200
- PH24358

Support client information to be propagated from a Db2 REST call

APAR PH24619 (June 2020) introduces the ability for Db2 native REST service users to optionally set Db2 client information registers by adding HTTP request header fields to a the Db2 native REST service request. In addition to the new HTTP request header field, this APAR also clarifies and enforces the Unicode code points that can be used for the Db2 REST HTTP request header client information fields.

Db2 native REST service users can override the current Db2 REST generated default values that are assigned to the CURRENT CLIENT_ACCTNG, CURRENT CLIENT_APPLNAME, CURRENT CLIENT_CORR_TOKEN, CURRENT CLIENT_USERID, and CURRENT CLIENT_WRKSTNNAME values. If the new header fields are not present, then the Db2 client information special registers will be populated with the default values.

For more information, see the following related topics:

- Management of REST service client information (Db2 REST services)
- Invoking a Db2 REST service (Db2 REST services)
- Creating a Db2 REST service (Db2 REST services)
- Dropping a Db2 REST service (Db2 REST services)
- Discovering all Db2 REST services (Db2 REST services)
- Discovering details of a Db2 REST service (Db2 REST services)
- PH24619

Improved reliability and contention relief for online migration

APAR PH10826 (May 2020) introduces the capability to identify applications that might be incompatible with catalog migrations and updates.

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

The new command and function are also available in Db2 11, and can be used for both system and non-system databases. For more information, see Identify applications that are incompatible with online migration (Db2 Installation and Migration).

For more information, see the following related topics:

- -DISPLAY BLOCKERS (Db2) (Db2 Commands)
- BLOCKING_THREADS (Db2 SQL)
- PH10826

Estimate space savings for Huffman compression using DSN1COMP

APAR PH19242 (April 2020) introduces the capability to estimate space savings for data in universal table spaces (UTS) when Huffman encoding is used with the hardware compression support in IBM z14 or newer hardware. It introduces a new optional COMPTYPE parameter for the DSN1COMP utility and also improves the format of the DSN1COMP compression report in message DSNI1940I.

If you run DSN1COMP for a UTS on IBM z14 or newer hardware, the default report now contains estimates for both fixed-length compression and Huffman encoding. You can also specify The new COMPTYPE parameter to specify which compression estimates to include in the compression report.

If you run DSN1COMP on IBM z13® or earlier hardware, or for a non-UTS table space, the default report also uses the new format, but it reports only the fixed-length compression estimates. However, if you specify COMPTYPE(ALL) or COMPTYPE(HUFFMAN) for a UTS on IBM z13 or earlier hardware, the compression report includes the Huffman compression estimates that you can expect after migration to IBM z14 or newer hardware.

For more information, see the following related topics:

- Syntax and options of the DSN1COMP control statement (Db2 Utilities)
- Sample DSN1COMP control statements (Db2 Utilities)
- DSN1940I (Db2 Messages)
- PH19242

Support for more granular collection of certain Db2 statistics traces

APAR PH18658 (April 2020) introduces more granular control for the collection of key Db2 statistics trace records, which were previously collected at fixed one-minute intervals. With faster processors, monitoring at a more granular interval can help in understanding the performance peak and analyzing slow down.

The new MAIN STATS TIME field is added on panel DSNTIPN for the STATIME_MAIN subsystem parameter. STATIME MAIN specifies the time interval, in seconds, between collection of certain interval-driven Db2 statistics trace records that are not collected at interval specified by the existing STATIME parameter. Previously these statistics were always collected at a fixed one-minute interval.

With this APAR, the STATIME_MAIN setting now controls the interval for collecting the following IFCIDs:

    0001 (System statistics)
    0002 (Db2 statistics)
    0202 (Buffer Pool Attributes)
    0225 (Storage statistics)

0230 (Group Buffer Pool Attributes)

0254 (Coupling Facility Cache statistics)

0369 (Wait and CPU time aggregated by connection type)

For more information, see the following related topics:

- Statistics trace (Db2 Performance)
- MAIN STATS TIME field (STATIME_MAIN subsystem parameter) (Db2 Installation and Migration)
- STATISTICS TIME field (STATIME subsystem parameter) (Db2 Installation and Migration)
- PH18658

Support for deleting remote services and related packages with FREE SERVICE commands

APAR PH21058 (March 2020) introduces the capability to specify a *location-name* value in the DSN FREE SERVICE subcommand, to remotely free a service that was previously copied, without requiring you to log in to the remote system that contains the service.

The location name must be defined in the SYSIBM.LOCATIONS table. If this table does not exist or the DBMS is not found, you receive an error message. If the location name is specified, the name of the local Db2 subsystem must be defined. If you omit *location-name,* the default is the local Db2 subsystem.

Although DSN BIND SERVICE COPY is the only way for a Db2 for z/OS requester to create a service on a Db2 for z/OS server, the remote FREE SERVICE support added by this APAR supports freeing any native REST service remotely, regardless of whether it was created with the CREATE SERVICE API or a DSN BIND SERVICE or BIND SERVICE COPY command.

For more information, see the following related topics:

- FREE SERVICE (DSN) (Db2 Commands)
- LOCATIONS catalog table (Db2 SQL)
- PH21058

Subsystem parameter default settings and ranges adjusted for Db2 best practices

APAR PH21370 (March 2020) updates the default settings, and in some cases the range of acceptable settings, for certain Db2 subsystem parameters. The changes are based on a review that revealed Db2 susbsytem parameters that accept settings or have default settings that do not comply with etablished Db2 best practices. Specifically, this APAR modifies the following Db2 subsystem parameters:

MINUTES/CHECKPOINT field (CHKFREQ and CHKMINS subsystem parameters) (Db2 Installation and Migration) – The default setting is changed from 5 minutes to 3 minutes. The documented range is change to 1 - 5 minutes. That is, Db2 now assumes that the setting is 5 minutes for any value greater than 5.

RO SWITCH TIME field (PCLOSET subsystem parameter) (Db2 Installation and Migration) – The default setting is changed from 10 minutes to 45 minutes.

RID POOL SIZE field (MAXRBLK subsystem parameter) (Db2 Installation and Migration) – The default setting is changed from 400,000 kilobytes to 1,000,000 kilobytes.

NPGTHRSH in macro DSN6SPRM (Db2 Installation and Migration) – The default setting is changed from 0 to 1.

INLISTP in macro DSN6SPRM (Db2 Installation and Migration) – The default setting is changed from 50 to 1000.

For more information, see the following related topics:

- "Subsystem parameter changes in Db2 12" on page 200
- PH21370

Support for creating artifacts for services that provision Db2 for z/OS data sharing groups

APAR PH19751 (March 2020) improves the Db2 install CLIST to support creating z/OSMF artifacts for provisioning and deprovisioning of Db2 data sharing groups, and for performing the following actions on provisioned Db2 data sharing group members:

- Start Db2
- Stop Db2
- Display DDF information
- Display data sharing group information
- Enable Optional Features: Db2 REST services, ODBC, and JDBC

A sample Db2 software service template is also now available for services that rapidly provision an *n*-way Db2 data sharing group (cluster), in IBM Cloud Provisioning and Management for z/OS (CP&M). The files (z/OSMF artifacts) of this sample software service template are compressed into the Db2ProvisionSystemDS.pax file, which you download onDb2ZTools/DevOps/Db2SystemServices on Github.

For more information, see the following related topics:

- Creating artifacts for provisioning Db2 by running the installation CLIST (Db2 Installation and Migration)
- Creating services for provisioning Db2 for z/OS (DBaaS) (Db2 Installation and Migration)
- Creating Db2 for z/OS data sharing group provisioning services
- PH19751

Add support for UNI_60 locale for the LOWER, TRANSLATE, and UPPER built-in functions

APAR PH19888 (March 2020) 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 6.0.0.

Prior to this APAR, Db2 allows the value 'UNI_SIMPLE' or 'UNI' as a locale 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. Locale '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. With this APAR, Db2 allows a new locale, UNI_60, which specifies that z/OS Unicode Conversion Services will be used with only "normal" casing capabilities and use Unicode Standard 6.0.0.

For more information, see the following related topics:

- LOWER (Db2 SQL)
- UPPER (Db2 SQL)
- TRANSLATE (Db2 SQL)
- PH19888

Serviceability improvements for insert algorithm 2

APAR PH18944 (March 2020)adds additional information to the DSNI055I and DSNI087I to improve serviceability for the insert algorithm 2 feature introduced in Db2 12. Specifically, it adds URID, RBA, and LRSN information for agents that enable or disable the use of insert algorithm 2.

For more information, see the following related topics:

- Insert algorithm 2 (Db2 Performance)
- DSNI055I (Db2 Messages)
- DSNI087I (Db2 Messages)
- PH18944

LOAD tolerates invalid PARALLEL specifications

With APAR PH19073 (March 2020), the PARALLEL option of the LOAD utility has several documented restrictions. Prior to PH19073, the LOAD utility failed with RC=8 when PARALLEL was specified in those restricted situations. With this APAR, the LOAD utility ignores PARALLEL in those situations and returns message DSNU073I to inform you that PARALLEL was ignored, and continues processing. For the specific list of situations where this option is ignored, see the description of the PARALLEL option in Syntax and options of the LOAD control statement (Db2 Utilities).

For more information, see the following related topics:

- PH19073

Automatic rebind operations attempt to reuse the access path from the current package

APAR PH15896 (February 2020) enforces APREUSE(WARN) and APREUSESOURCE(CURRENT) during autobind operations so that autobinds result in a comparable level of performance.

For Db2 10 and later packages, Db2 will attempt to reuse the access path from the current package copy. You don't need to change any system parameter settings, and this behavior is independent from the plan management and APREUSE that currently exist in the package.

For Db2 9 and earlier packages, the new behavior is not applied. To retain the access path from the current package copy, you must issue a REBIND command with APREUSE(WARN) and APREUSESOURCE(CURRENT).

For more information, see the following related topics:

- Reusing and comparing access paths at bind and rebind (Db2 Performance)
- Automatic rebinds (Db2 Application programming and SQL)
- APREUSE bind option (Db2 Commands)
- APREUSESOURCE bind option (Db2 Commands)
- PH19073

Ability to control the number of queued connections waiting for MONITOR THREADS profiles

APAR PH12041 (February 2020) introduces the following usability enhancements for monitoring threads and connections with profiles:

- IFCID READS capability for IFCID 0402 class 4 statistics records.
- New statistics for monitoring threads and connections in the IFCID 0402 class 4 record, including the following counters:
  - The current number of active threads
  - The current number of suspended threads
  - The highest number of threads since DDF start
  - The current number of connections
  - The highest number of connections since DDF start
- The ability to specify the maximum number of threads that can be queued waiting for a MONITOR THREADS profile. You can specify this limit in the ATTRIBUTE3 column of the DSN_PROFILE_ATTRIBUTES table. The ATTRIBUTE3 value must be a whole number that is either zero or less than or equal to the value of ATTRIBUTE2.

For more information, see the following related topics:

- Monitoring threads by using profile tables (Db2 Administration Guide)
- "IFCID changes in Db2 12" on page 193
- DSN_PROFILE_ATTRIBUTES profile table (Db2 Performance)
- PH12041

Ability to block certain Db2 utilities from updating replicated tables

Running certain utilities on tables or table spaces with replication enabled can cause the data in the replication target tables to be inconsistent until the tables are refreshed. APAR PH14363 (January 2020)) introduces a new subsystem parameter, UTILS_BLOCK_FOR_CDC, that prevents these utilities from updating replicated tables (those tables with the DATA CAPTURE CHANGES attribute) and therefore helps keep the replicated tables consistent.

For more information, see the following related topics:

- UTILS BLOCK FOR CDC field (UTILS_BLOCK_FOR_CDC subsystem parameter) (Db2 Installation and Migration)
- PH14363

Enable SECADM authority to grant BINDAGENT privilege to the binder

Prior to PH11309 (January 2020), the BINDAGENT privilege could only be granted between a specific grantor and grantee, preventing some automated processes from granting BINDAGENT to the binder when necessary. This APAR enables SECADM authority to grant BINDAGENT privileges to the binder, which allows the grantee to bind and rebind a plan or package by specifying any owner.

For more information, see the following related topics:

- GRANT (system privileges) (Db2 SQL)
- PH11309

Data set encryption key labels returned by ADMIN_DS_LIST

APAR PH12920 (January 2020) updates the ADMIN_DS_LIST stored procedure to return a flag that indicates whether the data set is encrypted with z/OS DFSMS data set encryption, and it also now returns the encryption key label. These values are returned in the new ENCRYPT_TYPE and ENCRYPT_KEY_LABEL columns in the result table. DFSMS APAR OA57544 is a prerequisite for this enhancement.

For more information, see the following related topics:

- Encrypting your data with z/OS DFSMS data set encryption (Managing Security)
- ADMIN_DS_LIST stored procedure (Db2 SQL)
- PH12920

Statistics profile options included in utility output

APAR PH11423 (January 2020) makes it easier to see the profile options that are being used when gathering statistics. When you run a utility with the USE PROFILE option, the profile options are listed in the SYSPRINT output data set. Prior to this APAR, if you wanted to see the profile options that are used, you first had to run the OPTIONS utility with the PREVIEW option to see the profile options. Then you had to execute the utility. With this APAR, you can see the profile options and execute the utility all in one step. The profile options continue to be displayed in the same message, DSNU1376I, as they were prior to this APAR by OPTIONS PREVIEW.

For more information, see the following related topics:

- Statistics profiles (Db2 Performance)
- DSNU1376I (Db2 Messages)
- PH11423

New-function APARs for Db2 12 in 2019

The following enhancements are introduced in the Db2 12 - initial release. That is, they are not controlled by the activation of function level 501 or higher. However, some of these enhancements might be controlled by the activation of function level 500, which is comparable to new-function mode in previous Db2 releases. They are listed roughly in order of availability, beginning with the most recent.

Validation for default buffer pools in the installation CLIST

APAR PH12117 (December 2019) introduces validation that default buffer pool names specified in the installation CLIST are defined and have a non-zero VPSIZE. To support this new capability the fields on panels DSNTIP1 and DSNTIP2 are reorganized with the following changes:

- All 4-KB buffer pool size fields are now specified on panel DSNTIP1, instead of being split across two panels.
- The default buffer pool names are now specified on panel DSNTIP2, instead of DSNTIP1.

DSNTIP2 also now validates that each specified default buffer pool exists and has a VPSIZE of greater than 0. For example, if you specify BP1 for the DEFAULT 4-KB BUFFER POOL FOR USER DATA field, the value of the BP1 field in DSNTIP1 must be 1 or greater. DSNTIP2 issues the existing DSNT566I message to report any validation errors.

For more information, see the following related topics:

- DSNTIP1: Buffer pool sizes panel 1 (Db2 Installation and Migration)
- DSNTIP2: Buffer pool sizes panel 2 (Db2 Installation and Migration)
- DSNT566I (Db2 Messages)
- PH12117

Support for deploying Db2 native REST services

APAR PH14688 (October 2019) provides updates to Db2 native REST services to support promotion of services from a development to production environments. It introduces the new options COPY, COPYVER, and OPTIONS to the existing DSN BIND SERVICE subcommand. These new options enable you to copy existing REST services to local or remote Db2 servers.

For more information, see the following related topics:

- BIND SERVICE (DSN) (Db2 Commands)
- COPY bind option (Db2 Commands)
- PH14688

Db2 12 function level 506 activation

Db2 12 function level 506 (PH16829 - October 2019) introduces support for new alternative names for existing built-in functions and support for implicitly dropping explicitly created table spaces.

For more information, see the following related topics:

- "Function level 506 (PH16829 - October 2019)" on page 81
- PH16829

IBM Db2 AI for z/OS 1.2.0.2

APARs PH14083 and PH16093 (October 2019) introduce support for Db2ZAI 1.2.0.2. Additional capabilities added by this release of Db2ZAI include the ability to manually lock in or unlock the access paths for static or dynamic SQL statements.

And Db2ZAI 1.2.0.2 now provides a RESTful API to schedule a system assessment in order to understand system health status or to detect a tuning opportunity.

Db2ZAI 1.2.0.2 also introduces a simplified installation process that significantly reduces the time to get your system up and running.

For more information, see the following related topics:

- IBM Db2 AI for z/OS 1.2.0.2 documentation
- PH14083
- PH16093

Ability to limit query acceleration processing to new accelerators

In a Db2 environment that includes a combination of V5 accelerators and V7 or later accelerators, APAR PH09995 (October 2019) enables you to limit EXPLAIN processing, accelerator modeling, and query execution to only V7 or later accelerators.

To enable this behavior, set the USE ONLY NEW ACCELERATOR_TYPE subsystem parameter option to YES. This new option is option 11 on the Acceleration Options Panel (DSNTIP8A).

For more information, see the following related topics:

- ACCELERATION OPTIONS field (QUERY_ACCEL_OPTIONS subsystem parameter) (Db2 Installation and Migration)
- PH09995

Ability to control workload balancing of accelerated queries based on accelerator version

In a Db2 environment that includes a mixture of IBM Db2 Analytics Accelerator for z/OS V5 and V7 accelerator servers, PH05477 (September 2019) enables the ability to control workload balancing such that queries can be accelerated to either a V5 accelerator or a V7 accelerator based on workload volume. Prior to PH05477, workload balancing was limited to a single version of an accelerator.

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

For more information, see the following related topics:

- ACCELERATION OPTIONS field (QUERY_ACCEL_OPTIONS subsystem parameter) (Db2 Installation and Migration)
- PH05477

IBM z15: Data privacy for diagnostics for buffer pool data in SVC dumps

With APAR PH15940 (September 2019), Db2 marks buffer pool data in SVC dumps as `sensitive=yes` for data privacy, and you can redact the data before sending the dump to IBM Support.

Enterprises have requirements to prevent customer personal or other sensitive information from being exposed to those who have no need to see the data. In the course of data processing, various types of system and application errors can require you to send diagnostic data to IBM Support, or other program vendors, for analysis and problem resolution. Among the types of diagnostic data usually collected for Db2 for z/OS, buffer pool data in dumps has the greatest exposure to containing sensitive data along with the system and or application data.

You can use the sample job SYS1.SAMPLIB(BLSJDPFD) to redact the buffer pool data that is tagged as `sensitive=yes` in SVC or stand-alone dumps before you send them to IBM Support or other vendors for problem analysis. However, you should retain original dump for the entire period that problem analysis is being conducted, in case IBM Support has specific questions about the redacted data.

You can use SYS1.SBLSCLI0(BLSXREDR)' and provide the input dump dataset name and optionally filtering ASID, to obtain a report about the pages which were marked as sensitive in a redacted dump.

For more information, see the following related topics:

- Redacting buffer pool data in SVC dumps for data privacy (Troubleshooting problems in Db2)
- Data Privacy for Diagnostics (DPfD)
- OA57633: IN SUPPORT OF OA57570
- PH15940

REPAIR utility enhancement to allow write of Db2 diagnostic log record

APAR PH11871 (September 2019) introduces a new REPAIR WRITELOG option to provide the ability for external applications to write a specific Db2 diagnostic log record to support the automatic refresh of replication target tables following a non-recoverable event.

For more information, see the following related topics:

- Syntax and options of the REPAIR control statement (Db2 Utilities)
- PH11871

Enhancements to insert algorithm 2

APAR PH02052 (August 2019) changes insert algorithm behavior so that when a storage or contention issue disables insert algorithm 2 (IAG2) and forces an INSERT statement to be run at the lower algorithm level, Db2 will automatically attempt to re-enable IAG2 at a set of predefined intervals. Before this APAR, when IAG2 became disabled due to a temporary condition, you needed to stop and restart the object to manually re-enable the insert algorithm, even when the situation that caused the failure had been resolved.

PH02052 also provides some usability and serviceability enhancements in the form of:

- Updates to message DSNI055I, which now provides more detailed explanations of the situations that can cause insert algorithm 2 to become disabled, as well as more details about potential actions that you can take in situations that might require user intervention - A new message, DSNI087I, which lets you know when insert algorithm 2 has been automatically re-enabled after a failure
- A new message, DSNI087I, which lets you know when insert algorithm 2 has been automatically re-enabled after a failure
- Two new IFCID 2 system level counters that can help you optimize insert algorithm 2 for your environment
- Enhanced diagnostic log records to indicate the successful re-enablement of insert algorithm 2
- Non-critical space map updates under s latch to improve performance
- The prevention of data page contention between concurrent insert algorithm 1 and insert algorithm 2 transactions
- The prevention of ABENDs such as DSNIBHUN:500C and DSNKINSL:5033, which result from insert algorithm 1 and insert algorithm 2 running concurrently on the same table space

For more information, see the following related topics:

- Insert algorithm 2 (Db2 Performance)
- DEFAULT INSERT ALGORITHM field (DEFAULT_INSERT_ALGORITHM subsystem parameter) (Db2 Installation and Migration)
- PH02052

Console message for inoperative packages

With APAR PH13550 (August 2019), Db2 now issues a DNST500I console message when a package becomes inoperative.

Before this APAR, there was no notification when a package became inoperative and needed to be rebound. This meant that when a package became inoperative, there could be a delay in informing the DBA of the issue with the result being an application outage.

For more information, see the following related topics:

- DSNT500I (Db2 Messages)
- Resource types (Db2 Messages)
- 00E30305 (Db2 Codes)
- PH13550

REORG SHRLEVEL CHANGE is compatible with LOAD RESUME YES SHRLEVEL CHANGE

With APAR PH11255 (August 2019), you can run a REORG TABLESPACE utility job with the SHRLEVEL CHANGE option and a LOAD utility job with the RESUME YES and SHRLEVEL CHANGE options concurrently on the same objects. Previously, these jobs were not compatible. REORG still requires exclusive control of the target objects in the last iteration in the LOG phase and in the SWITCH phase.

For more information, see the following related topics:

- Concurrency and compatibility for LOAD (Db2 Utilities)
- Concurrency and compatibility for REORG TABLESPACE (Db2 Utilities)
- PH11255

Hybrid Transactional Analytical Processing support for accelerated static queries

APAR PH14116 / PTF UI54294 (August 2019) introduces Db2 for z/OS support for Hybrid Transactional Analytical Processing (HTAP) of accelerated static queries. Db2 12 support for HTAP of accelerated dynamic queries was introduced by APAR PI91620 / PTF UI54293.

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

This HTAP functionality involves Db2 12 as the database for the real-time transactional data, IBM Db2 Analytics Accelerator for z/OS Version 5.1 or Version 7.1 as the data repository for the analytical data, and IBM InfoSphere Data Replication Change Data Capture for z/OS V10.2.1 to replicate the real-time transactional data from the Db2 database to the analytics accelerator. Additional PTFs are required for IBM Db2 Analytics Accelerator for z/OS and for IBM InfoSphere Data Replication Change Data Capture for z/OS.

For more information, see the following related topics:

- ACCELERATIONWAITFORDATA bind option (Db2 Commands)
- PH14116

New job DSNTIJPM premigration report 24: active log data sets larger than 4 GB

APAR PH11514 (July 2019) adds report 24 the to the premigration job DSNTIJPM. Report 24 identifies active log data sets greater than 4 GB in size. Before function level 500 is activated in Db2 12, the maximum supported active log data set size is 4 GB. Consequently, during migration to Db2 12, if the size of an active log data set in Db2 11 is greater than 4 GB, Db2 12 abnormally terminates during startup and issues message DSNJ158I. You can use the report to avoid this situation.

For more information, see the following related topics:

- Active log data sets storage requirements (Db2 Installation and Migration)
- Run premigration queries (DSNTIJPM) (Db2 Installation and Migration)
- DSNJ158I (Db2 Messages)
- PH11514

Support for sysplex group authentication

APAR PI94236 (July 2019) introduces support for sysplex group authentication, which is a security enhancement that enables the use of multi-factor authentication and RACF PassTickets in Db2 sysplex environments. Multi-factor authentication and RACF PassTickets are two distinct methods of providing additional authentication credentials at logon to verify a user's identity. Requiring an additional authentication credential ensures that a user's account cannot be compromised if one of their credentials is discovered.

Specifically, this APAR eliminates connection authentication failures that occur when DRDA sysplex workload balancing or a seamless failover client is being used. Prior to this APAR, connection authentication failures would occur because the drivers that perform sysplex workload balancing or seamless failover attempt to connect with multiple members of a data sharing group on behalf of the original client application connection. These extra connection attempts fail because the additional authentication credential expires or becomes void because it's valid only for the initial connection request and cannot be reused for subsequent connection requests.

For more information, see the following related topics:

- Enabling caching of MFA-based authentication credentials for clients with sysplex workload balancing (Managing Security)
- AUTH EXIT CACHE REFR (AUTHEXIT_CACHEREFRESH subsystem parameter) (Db2 Installation and Migration)
- DSNI055I (Db2 Messages)
- DSNI087I (Db2 Messages)
- PI94236

Asynchronous cross-invalidation of group buffer pools

z/OS introduced coupling facility asynchronous cross-invalidation in z/OS 2.2 and z/OS 2.3, with APAR OA54688. With APAR PH10577(June 2019), Db2 exploits this feature to provide improved elapsed time performance for writes of updated pages in a data sharing environment.

For more information, see the following related topics:

- Coupling facility structures (Db2 Installation and Migration)
- PH10577
- PH00193

Function level 505 activation

Db2 12 function level 505 (PH09191 - June 2019) introduces improved concurrency for rebind of executing packages, automatic page sample by default for RUNSTATS, improved DECFLOAT data type support, built-in functions for encryption using key labels, and temporal and archive transparency for WHEN clauses on triggers.

For more information, see the following related topics:

- "Function level 505 (PH09191 - June 2019)" on page 85
- PH09191

Loading multiple input data sets in a single job

In Db2 12 with APAR PI96136 (June 2019), you can load data from multiple data sets into the same target object in a single LOAD utility job. Previously, to load data from multiple data sets, you needed to either run multiple LOAD jobs or concatenate the data sets in the JCL. In some cases, concatenating the data sets is not possible. For example, if you run the LOAD utility by calling the DSNUTILU stored procedure, you cannot concatenate the data sets first. In Db2 12, you can now specify multiple input data sets in the INDDN option of the LOAD statement. You can specify either DD names or templates.

For more information, see the following related topics:

- Syntax and options of the LOAD control statement (Db2 Utilities)
- PI96136

Multiple captures on a single proxy for GDPS Continuous Availability with zero data loss

With APAR PI80787 (June 2019), the GDPS Continuous Availability with zero data loss solution provides disaster recovery with continuous application availability for enterprise businesses using the z/OS operating system. When Db2 is in a GDPS Continuous Availability with zero data loss environment, IFI applications issue READS calls for IFCID 0306 to a proxy data sharing group, and the proxy data sharing group captures log records from a source data sharing group.

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

For more information, see the following related topics:

- Reading complete log data for the GDPS Continuous Availability with zero data loss solution (Db2 Administration Guide)
- Dropping a Db2 REST service (Db2 REST services)
- BIND SERVICE (DSN) (Db2 Commands)
- FREE SERVICE (DSN) (Db2 Commands)
- PI80787

Creating artifacts for provisioning Db2 subsystems with the CLIST

APAR PH09857 (June 2019) introduces changes to support creating z/OSMF artifacts for provisioning and de-provisioning stand-alone Db2 subsystems (no data sharing) by running the installation CLIST. It requires z/OSMF Version 2.3 with APAR PH09130 or higher.

It also delivers a sample Db2 software service template to create services that rapidly provision from scratch one or multiple standalone Db2 subsystems, in IBM Cloud Provisioning and Management for z/OS. The z/OSMF artifacts of the sample software service template are compressed into the `Db2ProvisionSystemNonDS.pax` file, and this pax file is installed by default in the directory specified by the DDDEF created for SDSNASMP.

For more information, see the following related topics:

- Creating artifacts for provisioning Db2 by running the installation CLIST (Db2 Installation and Migration)
- Creating services for provisioning Db2 for z/OS (DBaaS) (Db2 Installation and Migration)
- PH09857

Db2 12 Function level 504 activation

Db2 12 function level 504 (PH07672 - April 2019) introduces new support for data with IBM zSystems hardware-based Huffman compression of Db2 data, the ability to prevent the creation of certain new deprecated objects types, new support for certain built-in functions by pass-through to IBM Db2 Analytics Accelerator, and new SQL syntax alternatives for certain special registers and NULL predicates.

For more information, see the following related topics:

- "Function level 504 (PH07672 - April 2019)" on page 89
- PH07672

Setting the client application compatibility property is optional when the client is Db2 for Linux®, UNIX, and Windows Version 11.1 Fix pack 1 or later

Prior to APAR PH08482 (April 2019), in a client-server environment in which the data server was Db2 for z/OS at function level V12R1M501 or later, setting the property value for client application compatibility (clientApplCompat or ClientApplCompat) was required. If that property value was not set, when a client or driver requested a connection with a data server at function level V12R1M501 or later, the request was rejected with SQLCODE -30025.

After the PTF for APAR PH08482 is applied, a Db2 for z/OS data server at function level V12R1M501 or later allows connections from a client or driver for which the client application compatibility property is not set. However, the data server continues to reject connections if the client, driver, or Db2 Connect Server gateway is at a level that does not support the APPLCOMPAT bind option value of the Db2 for z/OS package.

For more information, see the following related topics:

- Setting application compatibility levels for data server clients and drivers (Db2 Application programming and SQL)
- IBM Data Server Driver for JDBC and SQLJ properties for Db2 for z/OS (Db2 Application Programming for Java)
- PH08482

Allowing the PORT and SECPORT to be the same for static location aliases

Prior to APAR PH08188 (April 2019), Db2 restricted making the normal PORT and SECPORT the same for static location aliases. After the PTF for APAR PH08188 is applied, the PORT and SECPORT can be the same, which means that any TCPIP communication with Db2 as a server requires SSL connection setup. The rules for the main PORT and SECPORT, as well as any location alias PORT and SECPORT, will be the similar.

For more information, see the following related topics:

- -MODIFY DDF (Db2) (Db2 Commands)
- "Function level 504 (PH07672 - April 2019)" on page 89
- PH08188

Ability to delete only FlashCopy image copies

APAR PH04023 (April 2019) changes the MODIFY RECOVERY utility so that you can delete only FlashCopy image copies if equivalent sequential image copies exist. This new capability supports an efficient backup procedure that uses minimal disk space: create a FlashCopy image copy virtually instantaneously, create a sequential image copy from the FlashCopy image copy, and then delete the FlashCopy image copy. To delete this FlashCopy image copy, use the new FLASHCOPY ONLY option of MODIFY RECOVERY.

In previous Db2 releases, if you wanted to delete the FlashCopy image copy by using MODIFY RECOVERY, you had to also delete sequential image copies that met the same deletion criteria. Also, if you deleted the FlashCopy image copy by using mechanisms outside of Db2, Db2 did not know that copy was deleted and therefore did not have reliable information about backups.

For more information, see the following related topics:

- Backing up data efficiently by using FlashCopy image copies (Db2 Utilities)
- MODIFY RECOVERY (Db2 Utilities)
- PH04023

New job DSNTIJPM premigration report 23: SQL routines created in DB2 9

APAR PH07660 (April 2019) adds report 23 to the premigration job DSNTIJPM. Report 23 identifies native SQL procedures and compiled SQL scalar functions created in DB2 9. Such routines are subject to automatic binds after migration to Db2 12. You can use the report to regenerate these routines before migration to avoid performance problems that can result from too many concurrent automatic bind service tasks.

For more information, see the following related topics:

- Rebind old plans and packages in Db2 11 to avoid disruptive autobinds in Db2 12 (Db2 Installation and Migration)
- Run premigration queries (DSNTIJPM) (Db2 Installation and Migration)
- PH07660

Improve RUNSTATS performance by making page sampling the default behavior

One way to improve the performance of RUNSTATS is for the utility to use sampling instead of scanning every row and page. RUNSTATS can use either row sampling or page sampling. Page sampling is typically faster in both elapsed time and CPU time. APAR PH07220 (April 2019) introduces a new subsystem parameter, STATPGSAMP, that you can use to make page sampling the default behavior for RUNSTATS for universal table spaces. See "Function level 505 (PH09191 - June 2019)" on page 85.

In Db2 12 function level 505 and higher, the SYSTEM option, which means the same as NO in function level 504 and lower, is changed to mean the same as YES.

For more information, see the following related topics:

- PAGE-LEVEL SAMPLING field (STATPGSAMP subsystem parameter) (Db2 Installation and Migration)

- RUNSTATS TABLESPACE syntax and options (Db2 Utilities)
- "Function level 505 (PH09191 - June 2019)" on page 85
- PH07220

New environment variable to control the MQListener log size

APAR PH07230 (March 2019) lets you control the maximum size of the MQListener HFS log file using new environment variable MQLSNLWR.

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

For more information, see the following related topics:

- Environment variables for logging and tracing MQListener (Db2 Application programming and SQL)
- PH07230

IBM Db2 AI for z/OS 1.2.0

APAR PH07782 (March 2019) introduces support for Db2ZAI 1.2.0. Additional capabilities added by this release include the ability to detect Db2 system performance exceptions and to provide recommended actions for tuning which are based on your environment.

Another change is that Db2ZAI 1.2.0 now offers a simplified architecture, eliminating the need for a Linux environment.

For more information, see the following related topics:

- IBM Db2 AI for z/OS 1.2.0 documentation
- PH07782

IBM Db2 AI for z/OS1.1.0.1

APAR PH05323 (February 2019) introduces support for Db2ZAI 1.1.0.1. Additional capabilities added by this release include the ability to more accurately predict the benefit of parallel processing for static SQL (which can result in improved elapsed time and lower chargeable CPU) and the ability to determine whether sort processes can benefit from increased memory utilization.

Further, the ability to narrow the scope of operations to the package level has been added, allowing gradual exposure of applications to the benefits of machine learning and the ability to expand usage as confidence increases. Another change is that Db2ZAI1.1.0.1 will activate with a minimum Function Level of 500 provided that the required Db2 maintenance level (PH05323) is also available on all members of the data sharing group.

For more information, see the following related topics:

- IBM Db2 AI for z/OS 1.1.0.1 documentation
- PH05323

New in-memory resource limit table automatic refresh capability and enhancements to START RLIMIT

APAR PH06082 (February 2019)introduces an automatic refresh capability for in-memory resource limit tables. When you update the active set of resource limit tables, Db2 detects the resource limit changes and automatically refreshes the in-memory resource limit tables with these changes. You can now also issue a START RLIMIT command that specifies the current active set of resource limit tables without first issuing a STOP RLIMIT command.

Before this APAR, in-memory resource limit tables were not updated when you changed active resource limit tables until the resource limit facility was restarted. As a result, the changes did not take effect immediately for subsequent threads. Also, to start resource limit tables that were already active, you had to first stop the tables with the STOP RLIMIT command.

This APAR also adds the SCOPE(GROUP) keyword to the START RLIMIT command, providing the ability to start the resource limit facility for all members of a data sharing group. Previously, you could issue the START RLIMIT command for only a single member at a time.

For more information, see the following related topics:

- Specifying and changing resource limits (Db2 Performance)
- Starting and stopping resource limit tables (Db2 Performance)
- -START RLIMIT (Db2) (Db2 Commands)
- PH06082

Specify the default padding behavior for REORG TABLESPACE with UNLOAD EXTERNAL or DISCARD processing

APAR PH00317 (February 2019)introduces a new subsystem parameter that specifies the default NOPAD behavior for the REORG TABLESPACE utility with DISCARD or UNLOAD EXTERNAL processing. With this APAR, you can change the default padding behavior for the REORG TABLESPACE utility by setting the REORG_TS_NOPAD_DEFAULT subsystem parameter value. The default value is YES. The value of this subsystem parameter applies for REORG TABLESPACE utility control statements that omit the NOPAD keyword. However, the REORG TABLESPACE utility continues to honor user-specified NOPAD YES or NO keyword values in the utility control statement.

Before this APAR, if the REORG TABLESPACE utility control statement does not specify the NOPAD keyword the REORG utility always uses NOPAD YES for the default, with the result that the REORG never pads variable length column data in records discarded or unloaded, unless NOPAD NO is explicitly specified.

For more information, see the following related topics:

- REORG TS NOPAD DEFAULT (REORG_TS_NOPAD_DEFAULT subsystem parameter) (Db2 Installation and Migration)
- PH00317

Avoid external sorts for single-column column groups at the utility job level

With APAR PH03678 (February 2019, when you collect statistics for a single-column column group, avoiding external data sorts can improve utility performance. Previously, you could control the sort behavior by using the STATCLGRST subsystem parameter. STATCLGRST specifies the amount of memory a utility can use for sorting for single-column groups instead of invoking an external sort program. However, STATCLGRST is a global setting, and if the value is large, excessive amounts of memory might be allocated for all utility jobs. Now, you can control this behavior at the utility level by using the new keyword STATCLGMEMSRT. This keyword applies to RUNSTATS, LOAD, and REORG TABLEPSACE.

For more information, see the following related topics:

- STATISTICS COLGROUP DATA SORT STG LIMIT field (STATCLGSRT subsystem parameter) (Db2 Installation and Migration)
- LOAD (Db2 Utilities)
- RUNSTATS (Db2 Utilities)
- PH03678

Node.js support in Db2 for z/OS with the IBM_DB Node.js driver type

APAR PH05953 (January 2019) provides the necessary ODBC driver support to enable JavaScript web applications running on Node.js to establish connectivity to Db2 for z/OS using the IBM_DB Node.js driver in the z/OS UNIX System Services environment. Specifically, it enables wide (suffix-W) API for SQLGetDiagRec() to support Unicode wide character type

Before this APAR, the ODBC driver supports two entry points for each function that passes and accepts character string arguments: a generic API and a wide (suffix-W) API. Support for EBCDIC, ASCII, and UTF-8 encoded character string arguments is provided through APIs in the generic form.

However, support for UCS-2 encoded character string arguments is provided through APIs in wide form. This APAR adds support for the wide API form for SQLGetDiagRec().

For more information, see the following related topics:

- Node.js support in Db2 for z/OS (Db2 Programming for ODBC)
- Application encoding schemes and Db2 ODBC (Db2 Programming for ODBC)
- Suffix-W API function syntax (Db2 Programming for ODBC)
- PH05953

New-function APARs for Db2 12 in 2018

The following enhancements are introduced in the Db2 12 - initial release. That is, they are not controlled by the activation of function level 501 or higher. However, some of these enhancements might be controlled by the activation of function level 500, which is comparable to new-function mode in previous Db2 releases. They are listed roughly in order of availability, beginning with the most recent.

Db2 exploitation of asynchronous cross-invalidation for coupling facility cache structures

In a Db2 data sharing group, the elapsed time of writes to the group buffer pool can be quite long and unpredictable. This situation occurs especially in environments where the data sharing members are long distances from each other. z/OS APAR OA54688, which is available starting in z/OS 2.2, provides improved performance for transactions that involve multiple coupling facility cache requests by processing cross-invalidation requests asynchronously. With APAR PH05193 (December 2018), Db2 exploits this new z/OS capability to improve performance of group buffer pool write operations. Although data sharing groups whose members are far from each other can experience more substantial performance improvement, data sharing groups whose members are close together can also benefit.

For more information, see the following related topics:

- "IFCID changes in Db2 12" on page 193
- OA54688: NEW FUNCTION - ASYNCHRONOUS CROSS-INVALIDATION FOR COUPLING FACILITY CACHE STRUCTURES
- PI05193

Improved operational control for Db2 native REST services

APAR PI98649 (November 2018) provides a service update to the Db2 native REST services support, including both enhancements and defect fixes. It introduces three new Db2 commands, START RESTSVC, STOP RESTSVC, and DISPLAY RESTSVC for operational control of Db2 REST services and support for versioning of REST services.

With the STOP RESTSVC command, you can stop a service during maintenance so that invocation attempts will display an unavailable reply. After the maintenance is completed, the START RESTSVC command can be used to start the service again. The DISPLAY RESTSVC command displays the status of services, so you can see which services have been stopped.

The new Db2 REST service versioning support gives you the ability to define and use multiple versions of a REST service concurrently. Each version of a REST service can be created using a different SQL statement, different input and output parameters, and different bind options. This model provides the flexibility to create and deploy new versions of REST services to satisfy new application requirements without breaking existing applications.

For more information, see the following related topics:

- -DISPLAY RESTSVC (Db2) (Db2 Commands)
- -START RESTSVC (Db2) (Db2 Commands)
- -STOP RESTSVC (Db2) (Db2 Commands)
- REST services versioning (Db2 REST services)

- PI98649

New installation jobs for some DSNTIJMV job tasks

APAR PH02971 (November 2018) separates the various tasks in the DSNTIJMV installation job into several new jobs. This change addresses issues with separation of duties, provide more granular control over the tasks to complete for installation or migration, and to facilitate automation of installation, migration, and provisioning.

With APAR PH02971, the DSNTIJMV job now only defines Db2 to IRLM and MVS, and other tasks that were previously completed by DSNTIJMV are now moved to various other jobs, as shown in the following table.

Job name	Job purpose	Migration tasks	Installation tasks
DSNTIJMV (changed)	Defines Db2 to IRLM and MVS. Other actions previously completed by this job are moved to the new jobs.	Migration step 13: Define Db2 12 to IRLM and MVS: DSNTIJMV (Db2 Installation and Migration)	Installation step 2: Define Db2 to IRLM and MVS: DSNTIJMV (Db2 Installation and Migration)
DSNTIJMA (new)	Adds Db2 and IRLM address space startup procedures to the system procedure library.	Migration step 14: update Db2 address space start and language procedure definitions: DSNTIJMA, DSNTIJMW, DSNTIJMP, DSNTIJMJ (Db2 Installation and Migration)	Installation step 3: Define Db2 address space startup procedures, language procedures, and the Java WLM environment: DSNTIJMA, DSNTIJMW, DSNTIJMP, DSNTIJMJ (Db2 Installation and Migration)
DSNTIJMW (new)	Adds the address space startup procedures for the Db2-supplied WLM environments to the system procedure library.		
DSNTIJMJ (new)	Creates environment files for the Db2-supplied Java WLM environment.		
DSNTIJMP (new)	Copies the Db2 language procedures to the system procedure library.		

The specified jobs that the CLIST generates depend on the installation or migration scenario. For example, The installation CLIST generates DSNTIMW, DSNTIJMJ, and DSNTIJMP only for installation and migration of a standalone Db2 subsystem or the first member of a data sharing group. Job DSNTIJMA is always generated, but the procedures that it adds change depending on the scenario.

For more information, see PH02971.

New alternate format for timestamp values

APAR PH03263 (November 2018) introduces support for a new string representation for timestamp values. Expanding the allowable formats for timestamp values might decrease the amount of changes needed to port applications to Db2 12.

The new format has minus signs between the date and time portions and colons in the time portions of the string, similar to the following: `yyyy-mm-dd-hh:mm:ss`. This format is valid for both TIMESTAMP WITH TIME ZONE and TIMESTAMP WITHOUT TIME ZONE values. Refer to the description of TIMESTAMP *string-constant* in Datetime constants for details about all allowable formats for timestamp values.

For more information, see the following related topics:

- Datetime constants (Db2 SQL)
- CONNECT (Db2 command line processor) (Db2 Commands)
- The RACF secured signon PassTicket
- PI03263

Support for tracing native SQL routines

With PI44721 (October 2018) SQL procedures in Db2 provide the ability to trace native SQL routines. Those SQL procedures are:

DSN8.CREATE_DGTT
Creates a declared temporary table that is used by the sample trace procedures.

DSN8.ENABLE
Enables the SQL routine trace message buffer.

DSN8.GET_LINE
Returns a single line from the SQL routine trace message buffer. The returned data does not include an end-of-line character sequence.

DSN8.GET_LINES
Returns one or more lines from the SQL routine trace message buffer, and stores the lines in an array. The returned data does not include end-of-line character sequences.

DSN8.NEW_LINE
Writes an end-of-line character sequence to the SQL routine trace message buffer.

DSN8.PUT
Writes a line to the SQL routine trace message buffer, without an end-of-line character sequence.

DSN8.PUT_LINE
Writes a line to the SQL routine trace message buffer, with an end-of-line character sequence.

DSN8.DISABLE
Disables the SQL routine trace message buffer.

To define the SQL procedures and variables that are used by the SQL procedures, run job DSNTEJTR.

For more information, see the following related topics:

- DSN8.CREATE_DGTT stored procedure (Db2 SQL)
- DSN8.DISABLE stored procedure (Db2 SQL)
- DSN8.ENABLE stored procedure (Db2 SQL)
- DSN8.NEW_LINE stored procedure (Db2 SQL)
- DSN8.GET_LINE stored procedure (Db2 SQL)
- DSN8.GET_LINES stored procedure (Db2 SQL)
- DSN8.PUT stored procedure (Db2 SQL)
- DSN8.PUT_LINE stored procedure (Db2 SQL)
- Objects that are used by the sample trace stored procedures (Db2 SQL)
- PI44721

Function level 503 activation

Db2 12 function level 503 (PH00506 - October 2018) introduces replication of system-period temporal tables and generated expression columns, a change to temporal auditing support for temporal data, and a new console message that indicates when catalog levels or function levels change.

For more information, see the following related topics:

- "Function level 503 (PH00506 - October 2018)" on page 94
- PI00506

Allow ROTATE PARTITION for materialized query tables or tables having dependent materialized query tables

APAR PH00194 (October 2018) removes a restriction against rotating partitions when altering materialized query tables (MQTs) or tables with MQTs defined on them. Before this APAR, the ROTATE

PARTITION clause of ALTER TABLE statements was restricted if the tables being altered were MQTs or tables with dependent MQTs.

For more information, see the following related topics:

- Altering an existing materialized query table (Db2 Performance)
- ALTER TABLE (Db2 SQL)
- PI00194

Specify a DDF IPNAME

APAR PI99403 (October 2018)introduces support for specifying an IPNAME value when TCP/IP only communication is required. Before this APAR, the Db2 installation process did not support configuration of TCP/IP only communication.

You can specify the IPNAME value in the **DB2 TCP/IPNAME** field on panel DSNTIPR.

This APAR also reorganizes and renames the following panel fields:

- DB2 VTAM LUNAME field (Db2 Installation and Migration) replaces DB2 NETWORK LUNAME on panel DSNTIPR.
- DB2 VTAM PASSWORD field (Db2 Installation and Migration) replaces DB2 NETWORK PASSWORD on panel DSNTIPR.
- TCP/IP PORT replaces DRDA PORT, and this field is moved to panel DSNTIPR
- The following fields are also moved to panel DSNTIPR:
  - SECURE PORT field (Db2 Installation and Migration)
  - RESYNC PORT field (Db2 Installation and Migration)
- The following fields are moved to panel DSNTIP5:
  - DDF THREADS field (CMTSTAT subsystem parameter) (Db2 Installation and Migration)
  - MAX INACTIVE DBATS field (MAXTYPE1 subsystem parameter) (Db2 Installation and Migration)
  - IDLE THREAD TIMEOUT field (IDTHTOIN subsystem parameter) (Db2 Installation and Migration)
  - EXTENDED SECURITY field (EXTSEC subsystem parameter) (Db2 Installation and Migration)

For more information, see the following related topics:

- Specifying a generic LU name and IPNAME value for the data sharing group (Db2 Data Sharing Planning and Administration)
- DB2 TCP/IP IPNAME field (Db2 Installation and Migration)
- DSNJU003 (change log inventory) (Db2 Utilities)
- PI00506

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

APAR PH00335 (September 2018) introduces support for the Db2 command line processor to accept username and credential parameters when it is invoked to run SQL files. The credentials can be a password or RACF PassTicket and you can specify them in the new `-u` option flag when you start the Db2 command line processor.

Example: Specifying a user ID and password or RACF PassTicket to run an SQL file Suppose that you want to use the Db2 command line processor to connect to a server and your system administrator provides the following connection URL for the server: syszos1.abc.com:5021/ABCLOC1

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

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

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

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

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

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

For more information, see the following related topics:

- The Db2 command line processor (Db2 Commands)
- Start syntax for the Db2 command line processor (Db2 Commands)
- PI00335

Specify an HFS or zFS directory for z/OSMF artifacts

APAR PI97635 (July 2018) reduces the number of manual steps for installation and migration activities with z/OSMF, by introducing support for specifying an HFS or zFS directory for storing z/OSMF artifacts. Before this APAR, you must manually import the artifacts into HFS or zFS from MVS data sets that the Db2 installation CLIST generates.

You can specify the directory in the **PATH FOR Z/OSMF ARTIFACTS** field on panel DSNTIPM1, when you run the Db2 installation CLIST and specify YES in the **USE Z/OSMF WORKFLOW** field on panel DSNTIPA1. The user ID that runs the Db2 installation CLIST must have authority to create and write content to the specified directory.

For more information, see the following related topics:

- Installation panels for installing or migrating with z/OSMF (Db2 Installation and Migration)
- PI97635

UNLOAD utility uses REGISTER NO by default in Db2 12

APAR PI99075 (July 2018) changes the default behavior for the UNLOAD utility with SHRLEVEL CHANGE ISOLATION UR to REGISTER YES in Db2 12, which means that pages that are read by the UNLOAD utility are registered with the coupling facility in data sharing. In Db2 11 the UNLOAD utility always uses this behavior.

The new REGISTER NO behavior works as designed in Db2 12 to reduce data sharing overhead, in situations where the possible reduced currency of result data sets can be tolerated. To take advantage of the reduced data sharing overhead, specify REGISTER NO in utility jobs.

For more information, see the following related topics:

- Syntax and options of the UNLOAD control statement (Db2 Utilities)
- PI99075

Suppress SQL warning messages for OPEN and FETCH from SPUFI, DSNTEP2, DSNTEP4, and DSNTIAUL

APAR PI79053 (June 2018) introduces the QUIET option for the TOLWARN parameter of SPUFI and the DSNTEP2, DSNTEP4, and DSNTIAUL programs. QUIET is the same as YES, except that the program suppresses all SQL warning messages from OPEN or FETCH statements if the SQLCODE is 0 or greater.

For more information, see the following related topics:

- Controlling toleration of warnings in SPUFI (Db2 Application programming and SQL)
- DSNTEP2 and DSNTEP4 sample programs (Db2 Application programming and SQL)
- DSNTIAUL sample program (Db2 Application programming and SQL)

- PI79053

Db2 12 Function level 502 activation

Db2 12 function level 502 (PI95511 - May 2018) introduces management of key labels for z/OS DFSMS data set encryption and explicit casting of numeric values to GRAPHIC or VARGRAPHIC.

For more information, see the following related topics:

- "Function level 502 (PI95511 - May 2018)" on page 98
- PI95511

User-defined function with MODIFIES SQL DATA from a subselect

APAR PI93887 (April 2018) introduces support in Db2 12 for invoking a user-defined function that is defined with MODIFIES SQL DATA in a subselect. Previously, Db2 issued SQLCODE -740 for this case.

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

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

For more information, see the following related topics:

- ALTER FUNCTION (compiled SQL scalar) (Db2 SQL)
- ALTER FUNCTION (external) (Db2 SQL)
- CREATE FUNCTION (compiled SQL scalar) (Db2 SQL)
- CREATE FUNCTION (external scalar) (Db2 SQL)
- -740 (Db2 Codes)
- PI93887

Hybrid Transactional Analytical Processing support for accelerated queries

APAR PI91620 (March 2018) introduces Db2 12 support for Hybrid Transactional Analytical Processing (HTAP) of accelerated dynamic queries.

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

This HTAP functionality involves Db2 12 as the database for the real-time transactional data, IBM Db2 Analytics Accelerator for z/OS 5.1 as the data repository for the analytical data, and IBM InfoSphere Data Replication Change Data Capture for z/OS 10.2.1 to replicate the real-time transactional data from the Db2 database to the analytics accelerator. Additional PTFs are required for IBM Db2 Analytics Accelerator for z/OS and for IBM InfoSphere Data Replication Change Data Capture for z/OS.

For more information, see the following related topics:

- CURRENT QUERY ACCELERATION WAITFORDATA special register (Db2 SQL)
- SET CURRENT QUERY ACCELERATION WAITFORDATA (Db2 SQL)
- Making queries wait for incremental updates
- PI91620

LOAD REPLACE SHRLEVEL REFERENCE

APAR PI69095 (March 2018) introduces support for LOAD REPLACE SHRLEVEL REFERENCE, which enabled concurrent read application access to the target table while new data is being loaded into

a set of shadow data sets. A new SWITCH phase is introduced to the LOAD utility to switch access between the original and shadow data sets.

For more information, see the following related topics:

- Syntax and options of the LOAD control statement (Db2 Utilities)
- PI69095

Population of null attributes for objects created before function level V12R1M500

After you apply the PTF for APAR PI86123 (January 2018), the first of the following operations in function level V12R1M500 or higher populates any null attributes for objects created in function level V12R1M100 or earlier:

- ALTER TABLESPACE
- ALTER TABLE for limit key, adding partitions, or rotating partitions
- ALTER INDEX
- CREATE INDEX for partitioned indexes, if converting to table-controlled partitions
- DROP INDEX, if converting to table-controlled partitions

The purpose is to ensure that space-level attributes are populated in the Db2 catalog and OBDs and prevent subsequent REPAIR DBD DIAGNOSE operations on the object from reporting errors.

Important: Apply the PTF for APAR PH44109 to avoid unnecessary quiescing of dependent objects when Db2 populates these values.

- PI86123
- PH44109

HFS DRBM support for COBOL and PL/I

Before APAR PI88171 (January 2018), you can use the Db2 coprocessor and generate the DRBM as an HFS file when you invoke the C/C++ compiler from the z/OS UNIX System Services command line. This APAR introduces similar support for the COBOL Version 6.2 or newer and PL/I Version 5.2 or newer compilers.

The following example invokes the coprocessor to compile a sample COBOL program:

```
cob2 myprogram.cbl -c myprogram -dbrmlib -qsql
```

The following example invokes the coprocessor to compile a sample PL/I program:

```
pli -c -qpp=sql -qdbrmlib -qrent myprogram.pli
```

The COBOL and PL/I compilers pass the HFS filename for the DRBM by using the same convention as the C/C++ compiler to the coprocessor service. However, for COBOL and PL/I, the file name cannot be specified with the DBRMLIB option and the file name is always generated, as described in DRBMLIB.

For more information, see the following related topics:

- Processing SQL statements by using the Db2 coprocessor (Db2 Application programming and SQL)
- Binding a DBRM that is in an HFS file to a package or collection (Db2 Application programming and SQL)
- PI88171

Db2 Native REST services support update

APAR PI86868 (January 2018) introduces new BIND SERVICE and FREE SERVICE DSN subcommands for creating and dropping user-defined Db2 native REST services, and other enhancements and defect fixes. The new commands for creating and dropping Db2 user-defined REST services introduced by this APAR is similar to existing Db2 application package creation and deletion processes, is needed by Db2 DBAs.

Before this APAR, the Db2 native REST services support provided REST service manager APIs that can be used to create or drop Db2 user-defined REST services. These existing REST service manager APIs can be invoked from almost any REST client application or tooling, and support application developers very well. However, the requirement to use a REST client and the Db2 REST service manager APIs does not fit well with existing Db2 application deployment processes that are typically used by Db2 for z/OS database administrators (DBAs).

For more information, see the following related topics:

- Creating a Db2 REST service (Db2 REST services)
- Dropping a Db2 REST service (Db2 REST services)
- BIND SERVICE (DSN) (Db2 Commands)
- FREE SERVICE (DSN) (Db2 Commands)
- PI86868

Automatic remigration binds are removed in release coexistence and fallback

APAR PI87675 (January 2018) removes the possibility of disruptive repeating rebinds during release coexistence and fallback and remigration scenarios. The ABIND subsystem parameter is changed so that COEXIST behavior is used if YES is specified.

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

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

For more information, see the following related topics:

- Automatic binds in coexistence (Db2 Installation and Migration)
- AUTO BIND field (ABIND subsystem parameter) (Db2 Installation and Migration)
- PI87675

IBM zHyperLink: database synchronous read I/O exploitation

APAR PI82575 (January 2018) introduces support for performance improvements derived from IBM zHyperLink support. In a well-tuned environment, this feature can improve average response latencies for database page reads in online transactions.

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

For more information, see the following related topics:

- Read operations (Db2 Data Sharing Planning and Administration)
- DB2 zHyperLinks SCOPE field (ZHYPERLINK subsystem parameter) (Db2 Installation and Migration)
- PI82575

New-function APARs for Db2 12 in 2017

The following enhancements are introduced in the Db2 12 - initial release. That is, they are not controlled by the activation of function level 501 or higher. However, some of these enhancements might be controlled by the activation of function level 500, which is comparable to new-function mode in previous Db2 releases. They are listed roughly in order of availability, beginning with the most recent.

LOAD RESUME SUPPORT FOR INLINE COPY

To avoid unavailability of the table space or table space partition, the LOAD syntax has been enhanced with APAR PI81724 (December 2017) to allow the COPYDDN or RECOVERYDDN option to be specified

when RESUME YES is specified. A full image copy is taken at the end of LOAD processing, after potential BACKOUT processing.

For more information, see the following related topics:

- Syntax and options of the LOAD control statement (Db2 Utilities)
- PI81724

MQListener automatic restart after Db2 restart

APAR PI84698 (December 2017) introduces automatic restart of MQListener processing after Db2 is stopped and restarted, with the **-reconnectDB2** option of the **db2mqln1** and **db2mqln2 run** commands. Before this enhancement, you must restart the MQlistener after stopping and restarting Db2, which can cause service interruptions.

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

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

For more information, see the following related topics:

- Configuring MQListener tasks (Db2 Application programming and SQL)
- PI84698

New console message for remote client information

APAR PI89903 (December 2017) introduces message DSNL076I, which is issued to the z/OS system console for SQL statements that are routed to a server that does not support the requested function.

It identifies the remote client user and environment, the clientApplCompat value supplied by the remote client, and the name and APPLCOMPAT value of the associated Db2 package.

For more information, see the following related topics:

- DSNL076I (Db2 Messages)
- -30025 (Db2 Codes)
- PI89903

DSNTIJSG installation job no longer creates SYSIBM EXPLAIN tables

APAR PI86450 (December 2017) changes the DSNTIJSG installation job to remove CREATE statements for EXPLAIN tables with the SYSIBM qualifier. Db2 no longer uses these tables.

For more information, see the following related topics:

- Creating EXPLAIN tables (Db2 Performance)
- EXPLAIN tables (Db2 Performance)
- PI86450

CATMAINT utility LEVEL supports function level values

APAR PI88058 (December 2017) introduces support for Db2 12 function level values for the CATMAINT utility LEVEL option. When you run the CATMAINT utility to tailor the catalog for a new function level, you can specify either the target function level or the corresponding catalog level.

If you specify a function level value, Db2 determines the appropriate catalog level to tailor, and message DSNU777I indicates the result. If the catalog is already at the appropriate level, message DSNU766I indicates that no catalog update is required.

For more information, see the following related topics:

- Chapter 12, "Activating Db2 12 function levels," on page 281
- CATMAINT (Db2 Utilities)
- DSNU777I (Db2 Messages)
- DSNU776I (Db2 Messages)

- PI88058

DSNTIJUZ split into task-specific installation jobs

APAR PI85657 (November 2017) introduces several new jobs for various activities that were previously completed by the single DSNTIJUZ installation and migration job:

- Job DSNTIJUZ: define the Db2 data-only subsystem parameter module (Db2 Installation and Migration)
- Job DSNTIJUA: define data-only application defaults module (Db2 Installation and Migration)
- Job DSNTIJUM: define Db2 data-only offline message generator CCSID (DSNHMCID) module (Db2 Installation and Migration)
- Job DSNTIJUL: update DDF related BSDS information (Db2 Installation and Migration)

Before this APAR, the DSNTIJUZ job completes all of these activities, which often happen separately from the others, and for some only rarely. For example, updating the subsystem parameter module is a frequent task when new subsystem parameters are delivered in the service stream, or to update the default application compatibility (APPLCOMPAT) level. Continuous delivery and function levels in Db2 12 are likely to make this a more frequent activity. A similar more frequent activity with continuous delivery is rebuilding the Db2 application defaults module, such as to change the default SQL processing level. However, a rebuild of Db2 offline message generator CCSID module has never yet been required at migration, nor for applying Db2 service.

With the separate jobs, you can choose the jobs that you need for specific activities, without tailoring DSNTIJUZ to remove or disable unneeded job steps.

This APAR also modifies installation panels used by the z/OSMF installation and migration processes for Db2 12. In the following panels, the step for defining Db2 initialization parameters is split to use the new jobs.

For more information, see the following related topics:

- Sample workflow artifacts for installation and migration with z/OS Management Facility (Db2 Installation and Migration)
- Updating subsystem parameter and application default values (Db2 Installation and Migration)
- Chapter 14, "Updating Db2 initialization parameters for function level activation," on page 287
- Migration step 11: Define Db2 12 initialization parameters: DSNTIJUZ, DSNTIJUA, and DSNTIJUM (Db2 Installation and Migration)
- Installation step 6: Define Db2 initialization parameters: DSNTIJUZ, DSNTIJUA, DSNTIJUM, and DSNTIJUL (Db2 Installation and Migration)
- Defining Db2 initialization parameters for enabling data sharing (Db2 Installation and Migration)
- Defining Db2 initialization parameters for new data sharing members (Db2 Installation and Migration)
- Defining Db2 12 initialization parameters for migrating data sharing members (Db2 Installation and Migration)
- PI85657

Reduced cost for collecting Db2 frequency statistics

With APAR PI76730 (November 2017), your enterprise now has more control over performance when you collect frequency statistics for single-column column groups through RUNSTATS or inline statistics jobs. By adjusting the value of a new subsystem parameter, STATCLGSRT, you can potentially improve performance and reduce costs.

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

For more information, see the following related topics:

- Reducing the cost of collecting statistics (Db2 Performance)
- STATISTICS COLGROUP DATA SORT STG LIMIT field (STATCLGSRT subsystem parameter) (Db2 Installation and Migration)
- Syntax and options of the LOAD control statement (Db2 Utilities)
- Syntax and options of the REORG TABLESPACE control statement (Db2 Utilities)
- RUNSTATS TABLESPACE syntax and options (Db2 Utilities)
- PI76730

Changes to address problems after Db2 table definition changes

Db2 for z/OS database administrators often need to alter table definitions to support evolving requirements of applications. For table spaces with multiple tables, the following problems can occur:

- In certain situations, such as when a table is copied from one subsystem to another, the table definitions in the catalog might not match the data in the table space, which can lead to data integrity issues. This situation occurs most frequently when alterations are made to multiple tables in a table space.
- If a table space is at the maximum version level of 255, and at least one table in the table space is at version 0, you cannot recycle any table space versions. Until you recycle some table space versions, you cannot perform any more version-generating ALTER operations on tables in the table space.

APARs PI86880 and PI88940 (November 2017) provide a number of changes that improve your ability to synchronize the table definitions in the catalog with the data in a table.

Db2 processing changes:

- During UPDATE, INSERT, REORG, or LOAD processing on tables that have had no version-changing ALTER operations, Db2 adds information to the table spaces to make them self-describing. This action eliminates the need to rely on catalog and directory information to determine column definitions.
- After CREATE TABLE processing, and as soon as a table definition is complete, Db2 adds information to the table space to make the table self-describing. The new table has the version number of the table space.

Important: The previously mentioned Db2 processing changes can result in the addition of system pages to a table space, which increases the total size of the table space. You might also see an increase in the processing time for data definition statements, due to the allocation of additional pages in the table space.

Actions that you can take:

- You can run the REPAIR utility with the INSERTVERSIONPAGES and SETCURRENT options to insert missing metadata into a table space, or update table version numbers.

  INSERTVERSIONPAGES adds information to a table space or partition that has not had a version-changing ALTER operation, to make the table space self-describing. SETCURRENTVERSION, which must be specified with INSERTVERSIONPAGES and SHRLEVEL NONE, synchronizes the table version numbers for tables in a table space with the table space version number. The table space version number is the highest version number for any table in the table space. After you use SETCURRENTVERSION to update the versions of any tables that are at version 0, you can run the REORG and MODIFY RECOVERY utilities to recycle version numbers that are now unused.
- You can run the REORG or LOAD utility to insert missing metadata into a table space.

For more information, see the following related topics:

- Table space versions (Db2 Administration Guide)
- Copying tables from one subsystem to another (Db2 Utilities)
- Syntax and options of the REPAIR control statement (Db2 Utilities)

- PI86880
- PI88940

Support for collection of more accurate real storage usage statistics

With APAR PI78979 (August 2017), IFCID 0225 records contain new fields for collection of statistics about the number of discarded pages that are eligible for page steal. The IFCID 0225 record that is written at the end of the first one-minute statistics interval after midnight local time contains those statistics if the following conditions are true:

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

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

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

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

For more information, see the following related topics:

- MANAGE REAL STORAGE field (REALSTORAGE_MANAGEMENT subsystem parameter) (Db2 Installation and Migration)
- PI78979

New LOAD utility IGNORE options

APAR PI77159 (August 2017) introduces support for the following new IGNORE options for the LOAD utility.

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

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

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

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

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

For more information, see the following related topics:

- Syntax and options of the LOAD control statement (Db2 Utilities)
- PI77159

Real-time statistics collection of RUNSTATS-related columns starts at object creation

APAR PI79234 (July 2017)enhances real-time statistics collection for RUNSTATS-related columns in the SYSIBM.SYSTABLESPACESTATS and SYSIBM.SYSINDEXSPACESTATS catalog tables. With this APAR applied, the STATSLASTTIME column indicates the timestamp of the last time that the RUNSTATS utility is run on the table space or partition, or the time that table space or partition was created. The following columns are also updated to indicate the value since the object was created:

- STATSINSERTS
- STATSUPDATES

- STATSDELETES
- STATSMASSDELETES

For more information, see the following related topics:

- SYSTABLESPACESTATS catalog table (Db2 SQL)
- SYSINDEXSPACESTATS catalog table (Db2 SQL)
- PI79234

Explicitly-defined hidden ROWID columns

APAR PI77310 (June 2017) removes a restriction against explicitly defining a ROWID column with the IMPLICITLY HIDDEN attribute. Columns with the IMPLICITLY HIDDEN attribute are not returned for statements that do not explicitly list the column name in the select-clause. For example, the result for SELECT * does not include any implicitly hidden columns. To be included in the result, implicitly hidden columns must be explicitly specified in the select list.

For more information, see the following related topics:

- Row ID values (Db2 SQL)
- ROWID data type (Introduction to Db2 for z/OS)
- PI77310

Db2 Native Rest Services

APAR PI70652 (May 2017) introduces support for connecting RESTful web, mobile, and cloud applications to DB2 for z/OS by providing an environment for service, management, discovery, and invocation. Mobile and cloud application developers can use efficiently created, consumable, scalable, and RESTful services by using any REST client. They can use these services to securely interact with business-critical data and transactions, without special Db2 for z/OS expertise.

For more information, see the following related topics:

- Db2 REST services (Db2 REST services)
- PI70652

Db2 12 function level 501 activation

Db2 12 function level 501 (PI70535 - May 2017) introduces support for the LISTAGG built-in function.

For more information, see the following related topics:

- "Function level 501 (PI70535 - May 2017)" on page 101
- PI70535

ADMIN_INFO_SQL stored procedure enhanced to collect UDF information

APAR PI73268 (April 2017) enhances the ADMIN_INFO_SQL stored procedure and the DSNADMSB program to support the collection of information about user-defined functions (UDFs).

For more information, see the following related topics:

- Functions (Db2 SQL)
- ADMIN_INFO_SQL stored procedure (Db2 SQL)
- DSNADMSB (Db2 Utilities)
- PI73268

LOAD utility support for more date and time formats

With APAR PI69064 (November 2016), the LOAD utility supports a wider range of data and time formats. It introduces the following new DATE EXTERNAL (*date-format*) and TIME EXTERNAL (*time-format*) specifications.

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

Table 1. Values for date-format

date-format value	Format	Length
DATE_A	mm-dd-yyyy [1] on page 66	10 bytes
DATE_B	mm-dd-yy [1, 2]	8 bytes
DATE_C	yyyy-mm-dd [1] on page 66	10 bytes
DATE_D	yy-mm-dd [1, 2]	8 bytes
DATE_E	dd-mm-yyyy [1] on page 66	10 bytes
DATE_F	dd-mm-yy [1, 2]	8 bytes
DATE_G	yyyy-ddd [1] on page 66	8 bytes
DATE_H	yy-ddd [1, 2]	6 bytes
DATE_I	mmddyyyy	8 bytes
DATE_J	mmddyy [2] on page 66	6 bytes
DATE_K	yyyymmdd	8 bytes
DATE_L	yymmdd [2] on page 66	6 bytes
DATE_M	ddmmyyyy	8 bytes
DATE_N	ddmmyy [2] on page 66	6 bytes
DATE_O	yyyyddd	7 bytes
DATE_P	yyddd [2] on page 66	5 bytes

Notes:

1. If the format includes separators, the separator can be any single-byte character that can be converted to a single-byte EBCDIC character. However, if the input file has a delimited format, you cannot specify the same character that is used for a delimiter, including the COLDEL, CHARDEL, and DECPT delimiters. For more information about delimiter restrictions, see Loading delimited files (Db2 Utilities).

2. If a two-digit year is specified, it is expanded to a four-digit year. If the two-digit year is less than the sum of the two-digit current year plus 50, then the current century is used in the four-digit year. For example, assume that the current year is 2017. If the two-digit year is 67, 1967 is used. However, if the two-digit year is 66, 2066 is used.

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

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

Table 2. Values for time-format

time-format value	Format	Length
TIME_A	hh.mm.ss [1] on page 67	8 bytes
TIME_B	hh.mm [1] on page 67	5 bytes
TIME_C	hh.mm AM or hh.mm PM [1] on page 67	8 bytes
TIME_D	hhmmss	6 bytes

Table 2. Values for time-format (continued)

time-format value	Format	Length
TIME_E	hhmm	4 bytes

Notes:

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

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

For more information, see the following related topics:

- Syntax and options of the LOAD control statement (Db2 Utilities)
- PI69064

Chapter 2. Db2 12 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 12

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 12, see Part 2, "Adopting new capabilities in Db2 12 continuous delivery," on page 261.

Available function levels

The following function levels are available in Db2 12. They are listed in release order, beginning with the highest available function level.

Related information
Exploring IBM Db2 for z/OS Continuous Delivery (IBM Redpaper)

Function level 510 (V12R1M510) (PH33727 - April 2021)

Activating function level 510 (V12R1M510) enforces the completion of various pre-migration activities before migration to Db2 13, and is required before you can migrate a Db2 12 environment to Db2 13. It does not introduce any new features or capabilities, or catalog changes, in Db2 12.

Function level 510 (V12R1M510) verifies and enforces the following pre-migration conditions for migration to Db2 13 in your Db2 12 environment:

- All Db2 12 function levels are activated.
- All catalog updates for Db2 12 are applied, and the Db2 catalog is at the last Db2 12 catalog level.
- All recently used application packages are rebound recently enough to ensure that they are supported by Db2 13.

Important: Function level 510 (V12R1M510) does not include or verify that the fallback SPE (APAR PH37108 - April 2022), or other prerequisite maintenance for migration to Db2 13, is applied in Db2 12. For more information, see Required maintenance for Db2 13 installation or migration.

Activation of function level 510 (V12R1M510) fails if any of these conditions are not met.

Activation details for function level 510 (V12R1M510)

Function level 510 (V12R1M510) has the following activation details:

Enabling APAR:	PH33727

Minimum Db2 catalog level:	V12R1M509
Catalog changes:	None.
Application compatibility control:	None.

How to activate function level 510 (V12R1M510)

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 12 environment and continuous delivery in general, see Part 2, "Adopting new capabilities in Db2 12 continuous delivery," on page 261.

To activate function level 510 (V12R1M510) in preparation for migration to Db2 13, complete the following steps:

1. If Db2 12 is still at function level 100, activate function level 500 or 501 first. For more information, see Activating Db2 12 new function at migration (Db2 Installation and Migration).

2. Check that Db2 is at a sufficient code level by issuing a DISPLAY GROUP command, as described in Chapter 9, "Determining the Db2 code level, catalog level, and function level," on page 271. The DSN7100I message indicates the Db2 code level for under DB2 LVL in the member details. If DB2 LVL for any member is lower than 121510, apply the PTF for APAR PH33727 and any other required maintenance, so that all members indicate 121510.

   Tip: You can apply any PTF at any function level. It is best to run Db2 at this code level or higher for some time before you proceed with activating the function level. Db2 cannot run at any lower code level after you activate a function level, so you cannot remove any of the required PTFs after you activate a function level.

3. Run premigration job DSNTIJPE and resolve any release incompatibilities that it identifies.

4. Rebind any packages that are identified by the DSNTIJPE job. The job uses the following query to identify packages that were used in the last 18 months and were last bound or rebound in a release before Db2 11, and activation of function level 510 (V12R1M510) succeeds only if this query returns 0 rows.

```
SELECT *
  FROM SYSIBM.SYSPACKAGE
       WHERE LASTUSED >= DATE(DAYS(CURRENT DATE) - 548)
                     AND RELBOUND NOT IN ('P','Q','R')
                     AND VALID <> 'N'
                     AND OPERATIVE <> 'N';
```

   Until you complete this step, DISPLAY GROUP command output indicates HIGHEST POSSIBLE FUNCTION LEVEL (V12R1M509) or lower, depending on the catalog level.

   Important: If the DISABLE_EDMRTS subsystem parameter setting is YES in your Db2 12 environment, the SYSPACKAGE.LASTUSED column is not reliable. Function level 510 (V12R1M510) activation might succeed even though automatic rebinds remain likely to occur in Db2 13. To avoid such automatic binds from blocking the migration or requiring a fallback, adapt and follow the practices described in Rebind old plans and packages in Db2 11 to avoid disruptive autobinds in Db2 12 (Db2 Installation and Migration) as you prepare for migration to Db2 13.

5. Tailor the CATMAINT and function level activation jobs by running the installation CLIST:

   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 FUNCTON LEVEL ==> V12R1M510. 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).

6. Run the DSNTIJIC job to take an image copy of the Db2 catalog and directory.

7. Run the tailored DSNTIJTC job, or run the CATMAINT utility with LEVEL V12R1M510, 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.

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

```
-ACTIVATE FUNCTION LEVEL (V12R1M510) TEST
```

Db2 issues message DSNU757I to indicate the results. For more information, see Chapter 10, "Testing Db2 function level activation," on page 277.

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

```
-ACTIVATE FUNCTON LEVEL(V12R1M510)
```

Tip: If older packages run in Db2 12 after you activate function level 510, the HIGHEST POSSIBLE FUNCTION LEVEL value can indicate V12R1M509, even though the HIGHEST ACTIVATED FUNCTION LEVEL value is V12R1M510. In this case, you can still migrate to Db2 13 without any further action. However, possibly disruptive automatic rebinds might occur in Db2 13 unless you rebind these packages first. To determine which packages to rebind, run the queries in the DSNTIJPM or DSNTIJPE migration jobs.

Function level 510 incompatible changes

Function level 510 introduces no incompatible changes.

Function level 509 (PH33015 - February 2021)

Function level 509 (V12R1M509) introduces support for tamper-proof audit policies, high availability for accelerator-only tables, the specification of a compression algorithm at the object level, and a temporal RI enhancement to allow UPDATE or DELETE on a parent table.

Contents

"Support for tamper-proof audit policies" on page 72

"High availability for accelerator-only tables" on page 72

"Specify a compression algorithm at the table, table space, or partition level" on page 72

"Temporal RI allows UPDATE or DELETE on the parent table" on page 73

"Activation details for function level 509" on page 73

"How to activate function level 509" on page 73

"Function level 509 incompatible changes" on page 74

Finding function level 509 changes: To find the new and changed content for this function level, try searching for "FL 509" from any page. You'll see a list of new and changed topics for function level 509.

Throughout the Db2 12 information, when you see the link FL 509, the adjacent content was changed for function level 509, and you can click the link to see the page that you are currently reading.

Support for tamper-proof audit policies

Function level 509 introduces support for tamper-proof audit policies, which cannot be modified or stopped unless the user is authorized to access the Db2 audit policy profile by a z/OS security product, such as RACF, that is external to Db2. A z/OS system security administrator must perform a special task in the external security product to permit a Db2 user access to update, delete, or stop a tamper-proof audit policy. This new audit capability prevents users with advanced Db2 privileges from unnecessarily modifying or stopping the audit policy, minimizing the possibility of loss of audit information.

You can create a new tamper-proof audit policy by inserting an audit policy record into the SYSIBM.SYSAUDITPOLICIES catalog table with a DB2START value of 'T'. Any STOP TRACE commands on this record require additional RACF authorization in function level 509. UPDATE and DELETE statements on this record require additional RACF authorization in function level 509, regardless of the application compatibility level.

For more information, see:

- Updating tamper-proof audit policies (Managing Security)
- Creating and activating audit policies (Managing Security)
- -START TRACE (Db2) (Db2 Commands)
- -STOP TRACE (Db2) (Db2 Commands)

APAR PH31180 delivered the functional code to support tamper-proof audit policies.

High availability for accelerator-only tables

With function level 509, you can now define an accelerator-only table in more than one accelerator. This capability enhances accelerator-only tables with high availability and workload balancing, including the following features:

- Rerouting a query to an available accelerator if the target accelerator is not available
- Using the accelerator workload balancing algorithm to distribute queries based on the queue length on each accelerator

To create an accelerator-only table that is defined in multiple accelerators, specify a location alias that represents multiple accelerators in the IN ACCELERATOR clause of the CREATE TABLE statement. The accelerator-only table is defined in all accelerators that are associated with the location alias.

For more information, see:

- CREATE TABLE (Db2 SQL)
- Types of accelerator tables (Db2 Performance)
- Using an alias for an accelerator (Db2 Performance)

APAR PH30574 delivered the functional code for high availability for accelerator-only tables.

Specify a compression algorithm at the table, table space, or partition level

Function level 509 introduces the capability to explicitly specify either the fixed-length or Huffman compression algorithm at the table, table space, or partition level using the CREATE TABLE, CREATE TABLESPACE, and ALTER TABLESPACE statements. The Db2 catalog is updated to indicate the compression algorithm used for each object.

For more information, see:

- CREATE TABLE (Db2 SQL)
- CREATE TABLESPACE (Db2 SQL)
- ALTER TABLESPACE (Db2 SQL)
- Compressing your data (Db2 Performance)
- "Catalog changes in Db2 12" on page 179

APARs PH31729, PH31730, PH31731 delivered the functional code for specifying a compression algorithm at the table, table space, or partition level.

Temporal RI allows UPDATE or DELETE on the parent table

The UPDATE and DELETE statements are enhanced by removing restrictions related to the temporal referential integrity (RI) support that was introduced in the original Db2 12 release.

At application compatibility level V12R1M509 or higher, when an UPDATE statement with a FOR PORTION OF clause attempts to update the parent table in a temporal RI relationship, the update is allowed as long as the rules of temporal RI are not violated. Likewise, when a DELETE statement with a FOR PORTION OF clause attempts to delete from the parent table in a temporal RI relationship, the deletion is allowed, as long as the rules of temporal RI are not violated.

At any lower application compatibility level, such UPDATE or DELETE statements for a parent table in an RI relationship are restricted with SQLCODE -4736.

For more information, see:

- Temporal tables and data versioning (Db2 Administration Guide)
- "Support for temporal referential constraint" on page 230
- -4736 (Db2 Codes)

APAR PH30208 delivered the functional code for the temporal RI enhancement.

Activation details for function level 509

The following activation details apply to function level 509:

Enabling APAR:	PH33015
Minimum Db2 catalog level:	V12R1M509
Catalog changes:	A new COMPRESS_USED column is added in the SYSIBM.SYSTABLEPART catalog table, and new values 'F', and 'H' are added for the COMPRESS column in the following catalog tables: • SYSIBM.SYSTABLESPACE • SYSIBM.SYSTABLEPART
Application compatibility control:	The following new capabilities require that application runs at application compatibility level V12R1M509 or higher: • UPDATE or DELETE from the parent table in a temporal RI relationship.

How to activate function level 509

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 12 environment and continuous delivery in general, see Part 2, "Adopting new capabilities in Db2 12 continuous delivery," on page 261.

To activate Db2 12 function level 509, complete the following steps:

1. If Db2 12 is still at function level 100, activate function level 500 or 501 first. For more information, see Activating Db2 12 new function at migration (Db2 Installation and Migration).

2. Verify that any incompatible changes are resolved, including for any lower function levels not yet activated. See "Incompatible changes summary for function levels 501 and higher" on page 119.

3. Check that Db2 is at a sufficient code level by issuing a DISPLAY GROUP command, as described in Chapter 9, "Determining the Db2 code level, catalog level, and function level," on page 271. The

DSN7100I message indicates the Db2 code level for under DB2 LVL in the member details. If DB2 LVL for any member is lower than 121509, apply the PTF for APAR PH33015 and any other required maintenance, so that all members indicate 121509.

Tip: You can apply any PTF at any function level. It is best to run Db2 at this code level or higher for some time before you proceed with activating the function level. Db2 cannot run at any lower code level after you activate a function level, so you cannot remove any of the required PTFs after you activate a function level.

4. Tailor the CATMAINT and function level activation jobs by running the installation CLIST:

   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 ===> V12R1M509. 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).

5. Run the DSNTIJIC job to take an image copy of the Db2 catalog and directory.

6. Run the tailored DSNTIJTC job, or run the CATMAINT utility with LEVEL V12R1M509, 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.

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

```
-ACTIVATE FUNCTION LEVEL (V12R1M509) TEST
```

Db2 issues message DSNU757I to indicate the results. For more information, see Chapter 10, "Testing Db2 function level activation," on page 277.

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

```
-ACTIVATE FUNCTION LEVEL (V12R1M509)
```

9. 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 15, "Controlling the Db2 application compatibility level," on page 289.

   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.

Function level 509 incompatible changes

Function level 509 introduces no incompatible changes.

Function level 508 (PH29392 - October 2020)

Function level 508 (V12R1M508) introduces support for moving tables from deprecated multi-table simple and multi-table segmented table spaces to partition-by-growth universal table spaces (UTS), without taking an outage, to improve application and database operations and performance.

Contents

Finding function level 508 changes: To find the new and changed content for this function level, try searching for "FL 508" from any page. You'll see a list of new and changed topics for function level 508.

Throughout the Db2 12 information, when you see the link FL 508, the adjacent content was changed for function level 508, and you can click the link to see the page that you are currently reading.

Migration of multi-table table spaces to partition-by-growth universal table spaces (UTS)

Function level 508 enhances the ALTER TABLESPACE statement with a new MOVE TABLE option, which can be used to move a table from a source table space to a target table space. With this enhancement, you can now easily move tables from multi-table simple or multi-table segmented (non-UTS) table spaces, which are deprecated, to partition-by-growth UTSs. This new capability provides an efficient way to migrate existing data from deprecated table spaces to supported table spaces to take advantage of new functionality.

If the data sets of the source table space are already created, the MOVE TABLE operation becomes a pending definition change that must be materialized by running the REORG utility on the source table space.

For more information, see:

- Moving tables from multi-table table spaces to partition-by-growth table spaces (Db2 Administration Guide)
- ALTER TABLESPACE (Db2 SQL)
- Table space types and characteristics in Db2 for z/OS (Db2 Administration Guide)
- Materializing pending definition changes (Db2 Administration Guide)
- Accessing historical data from moved tables by using image copies (Db2 Administration Guide)

The following APARS delivered the functional code for the migration of multi-table table spaces to PBG UTS: PH27370, PH27658, PH27659, PH27660, PH27661, PH27662, and PH27663.

FTB support for non-unique indexes

With APAR PH30978 applied, function level 508 also extends the *fast index traversal* (often called "FTB" for "fast traverse blocks") feature introduced with the original release of Db2 12 to non-unique indexes.

FTBs provide valuable Db2 transaction performance improvements without the need for application changes. FTBs use a cache-friendly optimized in-memory structure to accelerate index-tree traversals, which can significantly reduce get page operations and CPU consumption. They are enabled by default in Db2 12 by the INDEX_MEMORY_CONTROL subsystem parameter setting and can take effect as soon as function level 100.

To enable use of FTBs for non-unique indexes, you set the value of the FTB_NON_UNIQUE_INDEX subsystem parameter to YES. The default value is NO.

For more information, see:

- Fast index traversal (Db2 Performance)
- Enabling or disabling fast index traversal at the index level (Db2 Performance)
- FTB NON UNIQUE INDEX field (FTB_NON_UNIQUE_INDEX subsystem parameter) (Db2 Installation and Migration)
- INDEX MEMORY CONTROL field (INDEX_MEMORY_CONTROL subsystem parameter) (Db2 Installation and Migration)
- PH30978

Activation details for function level 508

Function level 508 has the following activation details:

Enabling APAR:	APAR PH29392
Minimum Db2 catalog level:	V12R1M507
Catalog changes:	None
Application compatibility control:	Application compatibility level V12R1M508 or higher must be in effect for use of the new MOVE TABLE keyword of the ALTER TABLESPACE statement.

How to activate function level 508

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 12 environment and continuous delivery in general, see Part 2, "Adopting new capabilities in Db2 12 continuous delivery," on page 261.

To activate Db2 12 function level 508, complete the following steps:

1. If Db2 12 is still at function level 100, activate function level 500 or 501 first. For more information, see Activating Db2 12 new function at migration (Db2 Installation and Migration).

2. Verify that any incompatible changes are resolved, including for any lower function levels not yet activated. See "Incompatible changes summary for function levels 501 and higher" on page 119.

3. Check that Db2 is at a sufficient code level by issuing a DISPLAY GROUP command, as described in Chapter 9, "Determining the Db2 code level, catalog level, and function level," on page 271. The DSN7100I message indicates the Db2 code level for under DB2 LVL in the member details. If DB2 LVL for any member is lower than 121508, apply the PTF for APAR PH29392 and any other required maintenance, so that all members indicate 121508.

   Tip: You can apply any PTF at any function level. It is best to run Db2 at this code level or higher for some time before you proceed with activating the function level. Db2 cannot run at any lower code level after you activate a function level, so you cannot remove any of the required PTFs after you activate a function level.

4. Tailor the CATMAINT and function level activation jobs by running the installation CLIST:

   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 ===> V12R1M508. 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).

5. Run the DSNTIJIC job to take an image copy of the Db2 catalog and directory.

6. Run the tailored DSNTIJTC job, or run the CATMAINT utility with LEVEL V12R1M508, to update 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.

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

```
-ACTIVATE FUNCTION LEVEL (V12R1M508) TEST
```

Db2 issues message DSNU757I to indicate the results. For more information, see Chapter 10, "Testing Db2 function level activation," on page 277.

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

```
-ACTIVATE FUNCTION LEVEL (V12R1M508)
```

9. 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 15, "Controlling the Db2 application compatibility level," on page 289.

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.

Function level 508 incompatible changes

Function level 508 introduces no incompatible changes.

Function level 507 (PH24371 - June 2020)

Function level 507 (V12R1M507) introduces application granularity for locking limits, deletion of old statistics when using profiles, CREATE OR REPLACE PROCEDURE for external and native SQL procedures, and newly supported passthrough-only expressions with IBM Db2 Analytics Accelerator.

Contents

Application granularity for locking limits (NUMLKUS and NUMLKTS)

Two new built-in global variables are introduced to support application granularity for locking limits, which were previously specific only by the NUMLKUS and NUMLKTS subsystem parameters:

- SYSIBMADM.MAX_LOCKS_PER_TABLESPACE contains an integer value for the maximum number of page, row, or LOB locks that the application can hold simultaneously in a table space. If the application exceeds the maximum number of locks in a single table space, lock escalation occurs. MAX_LOCKS_PER_TABLESPACE corresponds to the existing NUMLKTS subsystem parameter.

- SYSIBMADM.MAX_LOCKS_PER_USER contains an integer value integer value that specifies the maximum number of page, row, or LOB locks that a single application can concurrently hold for all table spaces. The limit applies to all table spaces that are defined with the LOCKSIZE PAGE, LOCKSIZE ROW, or LOCKSIZE ANY options. MAX_LOCKS_PER_USER corresponds to the existing NUMLKUS subsystem parameter.

For more information, see:

- MAX_LOCKS_PER_TABLESPACE (Db2 SQL)
- Specifying the default maximum number of locks per table space (Db2 Performance)
- LOCKS PER TABLE(SPACE) field (NUMLKTS subsystem parameter) (Db2 Installation and Migration)
- MAX_LOCKS_PER_USER (Db2 SQL)
- Specifying the maximum number of locks that a single process can hold (Db2 Performance)
- LOCKS PER USER field (NUMLKUS subsystem parameter) (Db2 Installation and Migration)

APAR PH15342 delivered the functional code to support the application granularity for locking limits.

Deletion of old statistics when using profiles

When you run the RUNSTATS utility with the USE PROFILE option, Db2 collects only those statistics that are included in the specified profile. Beginning in function level 507, Db2 now also deletes existing statistics that are not part of the profile. All frequency, key cardinality, and histogram statistics that are not included in the profile are deleted from the catalog for the target object. This new behavior provides a way for you to remove stale distribution statistics without impacting concurrently running dynamic SQL.

This new capability also applies when profiles are used to gather inline statistics by the REORG TABLESPACE and LOAD utilities.

For more information, see:

- Statistics profiles (Db2 Performance)
- RUNSTATS TABLESPACE syntax and options (Db2 Utilities)

APAR PH16345 delivered the functional code to support the deletion of old statistics when using profiles.

CREATE OR REPLACE for procedures

You can now manage stored procedures in complex application environments more easily. The CREATE PROCEDURE statement (for external procedures and native SQL procedures) is extended to allow a new OR REPLACE clause. By adding the OR REPLACE clause to the CREATE PROCEDURE statement, you can reuse your original statement, make some changes to it, and reissue it to change the definition of an existing procedure. In many cases, this eliminates the need to first drop the existing procedure and then recreate it with the original CREATE statement that has been modified as needed.

Additionally, for native SQL procedures, you can use the OR REPLACE clause on a CREATE PROCEDURE statement in combination with a VERSION clause to replace an existing version of the procedure, or to add a new version of the procedure. When you reuse a CREATE statement with the OR REPLACE clause to replace an existing version or to add a new version of a native SQL procedure, the result is similar to using an ALTER PROCEDURE statement with the REPLACE VERSION or ADD VERSION clause. If the OR REPLACE clause is specified on a CREATE statement and a procedure with the specified name does not yet exist, the clause is ignored and a new procedure is still created.

- CREATE PROCEDURE (external) (Db2 SQL)
- CREATE PROCEDURE (SQL - native) (Db2 SQL)

APAR PH24324 delivered the functional code to support the new CREATE OR REPLACE PROCEDURE syntax.

Newly supported passthrough-only expressions with IBM Db2 Analytics Accelerator

The following built-in functions are now supported as passthrough-only expressions, which are passed through from Db2 for z/OS to IBM Db2 Analytics Accelerator:

- ADD_DAYS (Db2 SQL)
- BTRIM (Db2 SQL)
- DAYS_BETWEEN (Db2 SQL)
- NEXT_MONTH (Db2 SQL)
- Regression functions (REGR_AVGX, REGR_AVGY, REGR_COUNT, ...) (Db2 SQL)
- ROUND_TIMESTAMP (Db2 SQL) if invoked with a date expression

For passthrough-only expressions, Db2 for z/OS only verifies that the data types of the parameters are valid for the functions. The accelerator engine does all other function resolution processing and validation. For more information, see Accelerating queries with passthrough-only expressions.

APAR PH23042 delivered the functional code for the newly supported passthrough-only expressions.

Finding function level 507 changes: To find the new and changed content for this function level, try searching for "FL 507" from any page. You'll see a list of new and changed topics for function level 507.

Throughout the Db2 12 information, when you see the link FL 507, the adjacent content was changed for function level 507, and you can click the link to see the page that you are currently reading.

Activation details for function level 507

Function level 507 has the following activation details:

Enabling APAR:	PH24371.
Minimum Db2 catalog level:	V12R1M507
Catalog changes:	The following built-in global variables are introduced via a CATMAINT job: - MAX_LOCKS_PER_TABLE_SPACE - MAX_LOCKS_PER_USER
Application compatibility control:	Applications must run at application compatibility level V12R1M507 or higher to use the following new capabilities: - The SYSIBMADM.MAX_LOCKS_PER_TABLESPACE built-in global variable for overriding the NUMLKTS subsystem parameter. - The SYSIBMADM.MAX_LOCKS_PER_USER built-in global variable for overriding the NUMLKUS subsystem parameter. - CREATE PROCEDURE with the OR REPLACE and SPECIFIC clauses for external procedures and native SQL procedures. - Newly supported passthrough-only expressions.

How to activate function level 507

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 12 environment and

continuous delivery in general, see Part 2, "Adopting new capabilities in Db2 12 continuous delivery," on page 261.

To activate Db2 12 function level 507, complete the following steps:

1. If Db2 12 is still at function level 100, activate function level 500 or 501 first. For more information, see Activating Db2 12 new function at migration (Db2 Installation and Migration).

2. Verify that any incompatible changes are resolved, including for any lower function levels not yet activated. See "Incompatible changes summary for function levels 501 and higher" on page 119.

3. Check that Db2 is at a sufficient code level by issuing a DISPLAY GROUP command, as described in Chapter 9, "Determining the Db2 code level, catalog level, and function level," on page 271. The DSN7100I message indicates the Db2 code level for under DB2 LVL in the member details. If the highest function level that can be activated is lower than V12R1M507, apply the PTF for APAR PH24371 and any other required maintenance, so that all members indicate 121507 or higher for DB2 LVL in the DSN7100 message.

   Tip: You can apply any PTF at any function level. It is best to run Db2 at this code level or higher for some time before you proceed with activating the function level. Db2 cannot run at any lower code level after you activate a function level, so you cannot remove any of the required PTFs after you activate a function level.

4. Tailor the CATMAINT and function level activation jobs by running the installation CLIST:

   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 ===> V12R1M507. 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).

5. Run the DSNTIJIC job to take an image copy of the Db2 catalog and directory.

6. Run the tailored DSNTIJTC job, or run the CATMAINT utility with LEVEL V12R1M507, 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.

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

```
-ACTIVATE FUNCTION LEVEL (V12R1M507) TEST
```

   Db2 issues message DSNU757I to indicate the results. For more information, see Chapter 10, "Testing Db2 function level activation," on page 277.

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

```
-ACTIVATE FUNCTION LEVEL (V12R1M507)
```

9. 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 15, "Controlling the Db2 application compatibility level," on page 289.

   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.

Function level 507 incompatible changes

Function level 507 introduces no incompatible changes.

Function level 506 (PH16829 - October 2019)

Function level 506 (V12R1M506) introduces support for new alternative names for existing built-in functions and support for implicitly dropping explicitly created table spaces.

Contents

Finding function level 506 changes: To find the new and changed content for this function level, try searching for "FL 506" from any page. You'll see a list of new and changed topics for function level 506.

Throughout the Db2 12 information, when you see the link FL 506, the adjacent content was changed for function level 506, and you can click the link to see the page that you are currently reading.

Alternative function names support

To improve compatibility across the Db2 product family, the new alternative spellings shown in the following table are now supported as syntax alternatives for existing built-in functions that are already supported in Db2 for z/OS.

Table 3. Newly supported names for existing built-in functions

Newly supported alternative name	Existing equivalent function
CHAR_LENGTH	CHARACTER_LENGTH, which returns the length of its argument in the number of string units that are specified
COVAR_POP	COVARIANCE or COVAR, which return the population covariance of a set of number pairs

Table 3. Newly supported names for existing built-in functions (continued)

Newly supported alternative name	Existing equivalent function

HASH	HASH_MD5, HASH_SHA1, or HASH_SHA256, which return the result of applying a hash algorithm to an input argument, depending on the value specified for the second argument for the HASH function:
	
	0 (default)
	    HASH_MD5
	1
	    HASH_SHA1
	2
	    HASH_SHA256
	
	The HASH function returns a varying length (VARBINARY) value, unlike the existing functions, which return fixed length (BINARY) values.

POW	POWER, which returns the value of one argument raised to the power of a second argument
RANDOM	RANDOM, which returns a double precision floating-point random number
STRLEFT (Db2 SQL)	LEFT, which returns a string that consists of the specified number of leftmost bytes or the specified string units
STRPOS (Db2 SQL)	POSSTR, which returns the position of the first occurrence of an argument within another argument
STRRIGHT (Db2 SQL)	RIGHT, which returns a string that consists of the specified number of rightmost bytes or specified string units
TO_CLOB	CLOB, which returns a CLOB representation of the first argument
TO_TIMESTAMP (Db2 SQL)	TIMESTAMP_FORMAT, which returns a timestamp for a character string expression, using a specified format to interpret the string

APAR PH14712 delivered the functional code to support the new alternative names for existing built-in functions.

Support for implicitly dropping explicitly created UTS and LOB table spaces

Function level 506 introduces the following enhancements to dropping tables:

- If you drop a base table that resides in an explicitly created universal table space (UTS), Db2 implicitly drops the table space along with the table instead of returning an error in SQLCODE -669.
- If you drop a system-period temporal table or an archive-enabled table, Db2 also drops the associated history table or archive table. If the history table or archive table resides in an explicitly created UTS, Db2 implicitly drops the table space along with the table instead of returning an error in SQLCODE -669.
- If you drop an auxiliary table that resides in an explicitly created LOB table space, Db2 implicitly drops the table space along with the table. Previously, the LOB table space remained as an empty table space. An auxiliary table is dropped when you take one of the following actions:
  - Issue the DROP TABLE statement on the auxiliary table
  - Drop the associated base table
  - Drop an associated LOB column

– Remove an associated trailing empty partition when running the REORG utility on a partition-by-growth table space

For more information, see DROP (Db2 SQL).

The new REORG utility behavior takes effect immediately after the activation of function level 506 or higher. For the new SQL statement behavior to take effect, the application compatibility level must be set to V12R1M506 or higher.

APAR PH14452 introduced the functional code to support implicitly dropping explicitly created table spaces.

Activation details for function level 506

Function level 506 has the following activation details:

Enabling APAR:	PH16829.
Minimum Db2 catalog level:	V12R1M505
Catalog changes:	None.
Application compatibility control:	Applications must run at application compatibility level V12R1M506 or higher to use the following new capabilities: • Newly supported names for existing built-in functions. • DROP statement enhancements for implicitly dropping explicitly created table spaces.

How to activate function level 506

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 12 environment and continuous delivery in general, see Part 2, "Adopting new capabilities in Db2 12 continuous delivery," on page 261.

To activate Db2 12 function level 506, complete the following steps:

1. If Db2 12 is still at function level 100, activate function level 500 or 501 first. For more information, see Activating Db2 12 new function at migration (Db2 Installation and Migration).

2. Verify that any incompatible changes are resolved, including for any lower function levels not yet activated. See "Incompatible changes summary for function levels 501 and higher" on page 119.

3. Check that Db2 is at a sufficient code level by issuing a DISPLAY GROUP command, as described in Chapter 9, "Determining the Db2 code level, catalog level, and function level," on page 271. The DSN7100I message indicates the Db2 code level for under DB2 LVL in the member details. If DB2 LVL for any member is lower than 121506, apply the PTF for APAR PH16829 and any other required maintenance, so that all members indicate 121506.

   Tip: You can apply any PTF at any function level. It is best to run Db2 at this code level or higher for some time before you proceed with activating the function level. Db2 cannot run at any lower code level after you activate a function level, so you cannot remove any of the required PTFs after you activate a function level.

4. Tailor the CATMAINT and function level activation jobs by running the installation CLIST:

   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 ===> V12R1M506. 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).

5. Run the DSNTIJIC job to take an image copy of the Db2 catalog and directory.

6. Run the tailored DSNTIJTC job, or run the CATMAINT utility with LEVEL V12R1M506, to update the 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.

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

```
-ACTIVATE FUNCTION LEVEL (V12R1M506) TEST
```

Db2 issues message DSNU757I to indicate the results. For more information, see Chapter 10, "Testing Db2 function level activation," on page 277.

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

```
-ACTIVATE FUNCTION LEVEL (V12R1M506)
```

9. 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 15, "Controlling the Db2 application compatibility level," on page 289.

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.

Function level 506 incompatible changes

Function level 506 introduces the following incompatible changes:

Changes in explicitly created table space behavior

When the application compatibility level is set to V12R1M506 or higher, the following changes become effective:

- Dropping a table that resides in an explicitly created universal table space no longer returns an error. Instead, the table space is implicitly dropped.
- Dropping an auxiliary table that resides in an explicitly created LOB table space no longer leaves the LOB table space in the database. Instead, the table space is implicitly dropped.

Actions required:

- Any existing applications that use the DROP TABLESPACE statement to drop a universal table space with a table can now be changed to use the DROP TABLE statement instead.
- If an auxiliary table is dropped, any applications or tools that expect the LOB table space to remain for reuse must be modified accordingly.

Function level 505 (PH09191 - June 2019)

Function level 505 (V12R1M505) introduces improved concurrency for rebind of executing packages, automatic page sample by default for RUNSTATS, improved DECFLOAT data type support, built-in functions for encryption using key labels, and temporal and archive transparency for WHEN clauses on triggers.

Contents

Finding function level 505 changes: To find the new and changed content for this function level, try searching for "FL 505" from any page. You'll see a list of new and changed topics for function level 505.

Throughout the Db2 12 information, when you see the link FL 505, the adjacent content was changed for function level 505, and you can click the link to see the page that you are currently reading.

Improved Hybrid Transactional Analytical Processing performance

Performance improvements address the perceived latency between Db2 for z/OS and IBM Db2 Analytics Accelerator in Hybrid Transactional Analytical Processing (HTAP) environments. APAR PI98630 delivered the functional code.

Rebind phase-in for packages that are being used for execution

Function level 505 introduces support for rebind phase-in, an enhancement that allows Db2 to rebind a package concurrently with execution of the package. To improve concurrency, a rebind operation now creates a new copy of the package. When the rebind operation finishes, new threads can use the new package copy immediately, and existing threads can continue to use the copy that was in use prior to the rebind (the phased-out copy) without disruption.

Db2 waits for the duration specified by the IRLMRWT subsystem parameter before it creates package copies for the rebind phase-in operation.

This enhancement also enables switching to previous access paths and runtime structures gradually (the switch to the previous access path is phased-in), which allows regression recovery without incurring an application outage.

For more information, see the following topics:

• Phase-in of package rebinds (Db2 Application programming and SQL)
• REBIND PACKAGE subcommand (DSN) (Db2 Commands)

APAR PH12186 delivered the functional code for new rebind phase-in capability.

Tip: APAR PH28693 (January 2021) improves concurrency for REBIND commands in Db2 12 at function level 505 or higher. With this APAR, a REBIND command now always obtains a U lock, allowing subsequent transactions that are executing a package to run in parallel. For more information, see "Improved transaction execution times and concurrency for REBIND PACKAGE" on page 28

New built-in functions for encryption and decryption with key labels

Function level 505 introduces new built-in encrypt and decrypt data key functions to simplify data protection. The new built-in function ENCRYPT_DATAKEY (Db2 SQL) converts a block of plain text to a block of encrypted text using a specified algorithm and key label. The new built-in DECRYPT_DATAKEY functions return the block of encrypted text from ENCRYPT_DATAKEY as a block of the datatype specified by the function used.

Depending on your security configuration, Db2 might create an ACEE for the primary authorization ID to check for key label access during execution of ENCRYPT_DATAKEY or DECRYPT_DATAKEY_*type* built-in functions. If you are using external security for authorization, be aware that Db2 might also start providing this ACEE for other subsequent authorization checks, which can result in a different return codes from the exit. For more information, see When Db2 creates an ACEE (RACF Access Control Module Guide) and Example 5: Deferring to Db2 (missing ACEE) (RACF Access Control Module Guide).

APAR PH09506 delivered the functional code for the new built-in encryption and decryption functions.

Improved support for DECFLOAT columns

Function level 505 introduces improvements to the support for DECFLOAT columns to improve performance of mission critical applications. These improvements include the ability to specify DECFLOAT columns in an index and as a key in a primary or unique key.

For more information about using DECFLOAT columns in primary and unique keys, see CREATE TABLE (Db2 SQL) and ALTER TABLE (Db2 SQL).

For more information about using DECFLOAT columns in indexes, including changes to the maximum length of an index key, see CREATE INDEX (Db2 SQL) and ALTER INDEX (Db2 SQL).

APAR PH09797 delivered the functional code for the improved support of DECFLOAT columns.

Improved RUNSTATS performance with automatic page sampling by default

Function level 505 changes the meaning of the default value of the STATPGSAMP subsystem parameter to mean the same as YES, which means that RUNSTATS uses page-level sampling by default for universal table spaces. STATPGSAMP was introduced in Db2 12 by APAR PH07220. The default value is SYSTEM.

In function level 504 and lower, SYSTEM has the same meaning as NO. That is, RUNSTATS does not use page sampling by default. Any sampling is determined by the options that the RUNSTATS statement specifies.

As a result of this enhancement, RUNSTATS no longer updates real-time statistics by default. For possible actions to take, see "Function level 505 incompatible changes" on page 88.

For more information, see PAGE-LEVEL SAMPLING field (STATPGSAMP subsystem parameter) (Db2 Installation and Migration) and RUNSTATS TABLESPACE syntax and options (Db2 Utilities).

APAR PH07220 delivered the functional code for the new page default page sampling behavior.

Temporal and archive transparency support for WHEN clause on triggers

Function level 505 allows system-period temporal tables and archive-enabled tables to be referenced in WHEN clauses of both basic and advanced triggers, regardless of the settings of the SYSTIMESENSITIVE and ARCHIVESENSITIVE bind options. Time machine and transparent archive data retrieval are fully supported in the WHEN clause.

APAR PH09794 delivered the functional code for the temporal and archive transparency support for WHEN clause on triggers.

Activation details for function level 505

Function level 505 has the following activation details:

Enabling APAR:	PH09191.
Minimum Db2 catalog level:	V12R1M505 [1] on page 87
Catalog level V12R1M505 changes:	A new COPYID column is added in the following tables: • SYSIBM.SYSPACKDEP • SYSIBM.SYSPACKSTMT
Application compatibility control:	Applications must run at application compatibility level V12R1M505 or higher to use the following new capabilities: • All built-in functions described in "New built-in functions for encryption and decryption with key labels" on page 86. • Creating a new index with a DECFLOAT column as a key, or adding a DECFLOAT column to an existing index. • Defining a primary or unique constraint with a DECFLOAT column. • Temporal and archive transparency support for WHEN clause on triggers. .

Notes:

1. Before tailoring the Db2 catalog for catalog level V12R1M505, apply the PTFs for the following APARs: PH19720 and PH15258.

How to activate function level 505

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 12 environment and continuous delivery in general, see Part 2, "Adopting new capabilities in Db2 12 continuous delivery," on page 261.

To activate Db2 12 function level 505, complete the following steps:

1. If Db2 12 is still at function level 100, activate function level 500 or 501 first. For more information, see Activating Db2 12 new function at migration (Db2 Installation and Migration).

2. Verify that any incompatible changes are resolved, including for any lower function levels not yet activated. See "Incompatible changes summary for function levels 501 and higher" on page 119.

3. Check that Db2 is at a sufficient code level by issuing a DISPLAY GROUP command, as described in Chapter 9, "Determining the Db2 code level, catalog level, and function level," on page 271. The DSN7100I message indicates the Db2 code level for under DB2 LVL in the member details. If DB2 LVL for any member is lower than 121505, apply the PTF for APAR PH09191 and any other required maintenance, so that all members indicate 121505.

   Tip: You can apply any PTF at any function level. It is best to run Db2 at this code level or higher for some time before you proceed with activating the function level. Db2 cannot run at any lower code level after you activate a function level, so you cannot remove any of the required PTFs after you activate a function level.

4. Verify that the PTFs for the following APARs are applied: PH19720 and PH15258.

5. Tailor the CATMAINT and function level activation jobs by running the installation CLIST:

   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 ===> V12R1M505. 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).

6. Run the DSNTIJIC job to take an image copy of the Db2 catalog and directory.

7. Run the DSNTIJTC job, or run the CATMAINT utility with LEVEL V12R1M505, to update 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.

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

```
-ACTIVATE FUNCTION LEVEL (V12R1M505) TEST
```

Db2 issues message DSNU757I to indicate the results. For more information, see Chapter 10, "Testing Db2 function level activation," on page 277.

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

```
-ACTIVATE FUNCTION LEVEL (V12R1M505)
```

10. 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 15, "Controlling the Db2 application compatibility level," on page 289.

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.

Function level 505 incompatible changes

Function level 505 introduces the following incompatible changes:

RUNSTATS does not update real-time statistics by default

Starting in function level 505, the RUNSTATS utility no longer updates real-time statistics (RTS) by default.

Action required: If you want RUNSTATS to update RTS, take one of the following actions:

- In the RUNSTATS statement, specify TABLESAMPLE SYSTEM NONE and do not specify SAMPLE.
- Set the STATPGSAMP subsystem parameter to NO and do not specify SAMPLE in the RUNSTATS statement.

Newly supported user-defined functions

If your environment contains existing user-defined functions with the same names and signatures as the new built-in functions, applications with unqualified references to the existing user-defined functions might start invoking the built-in function instead of the user-defined function in certain situations.

Action required: For information about avoiding such situations, see Ensuring that Db2 executes the intended user-defined function (Db2 Application programming and SQL).

Function level 504 (PH07672 - April 2019)

Function level 504 (V12R1M504) introduces new support for data with IBM zSystems hardware-based Huffman compression of Db2 data, the ability to prevent the creation of certain new deprecated objects types, new support for certain built-in functions by pass-through to IBM Db2 Analytics Accelerator, and new SQL syntax alternatives for certain special registers and NULL predicates.

Contents

Finding function level 504 changes: To find the new and changed content for this function level, try searching for "FL 504" from any page. You'll see a list of new and changed topics for function level 504.

Throughout the Db2 12 information, when you see the link FL 504, the adjacent content was changed for function level 504, and you can click the link to see the page that you are currently reading.

IBM z14 Huffman compression

Function level 504 introduces support for compression of Db2 data with IBM zSystems hardware-based entropy encoding (Huffman) compression with the IBM z14 Compression Coprocessor (CMPSC).

The TS_COMPRESSION_TYPE subsystem parameter controls the compression method for Db2 subsystems on IBM z14 hardware with Huffman compression enabled. Huffman compression is only available for data in universal table spaces, regardless of the setting of the TS_COMPRESSION_TYPE subsystem parameter.

For more information, see TS COMPRESSION TYPE field (TS_COMPRESSION_TYPE subsystem parameter) (Db2 Installation and Migration) and Using Huffman compression to compress your data (Db2 Performance).

APAR PH04424 delivered the functional code that supports IBM z14 Huffman compression.

Prevent creation of new deprecated objects

Certain Db2 capabilities are deprecated. That is, although they remain supported, they have been replaced by other capabilities, their continued use is not recommended, and support for them is likely to be removed in the future. For a full list of such capabilities in Db2 12, see "Deprecated function in Db2 12" on page 220.

Function level 504 introduces capability to prevent the creation of certain deprecated objects in your Db2 for z/OS environments. Starting in function level 504, SQL statement in packages that are bound with APPLCOMPAT(V12R1M504) or higher, or with comparable SQL options in effect, are prevented from creating following types of deprecated objects:

Synonyms

The CREATE SYNONYM statement results in an error. Use aliases instead of synonyms. For more information about aliases, see Aliases (Introduction to Db2 for z/OS) and CREATE ALIAS (Db2 SQL).

Segmented (non-UTS) or partitioned (non-UTS) table spaces

The result of a CREATE TABLESPACE statement that specifies the NUMPARTS clause but no MAXPARTITIONS clause is a partition-by-range table space. Otherwise, the result is a partition-by-growth table space.

If no SEGSIZE clause is specified, Db2 uses the DPSEGSZ subsystem parameter value. For more information, see DEFAULT PARTITION SEGSIZE field (DPSEGSZ subsystem parameter) (Db2 Installation and Migration).

Table 4. CREATE TABLESPACE clauses for specifying table space types, by application compatibility level.

Table space type	APPLCOMPAT(V12R1M504) and higher	APPLCOMPAT(V12R1M503) and lower
Partition-by-growth	Any of the following combinations: • MAXPARTITIONS and NUMPARTS • MAXPARTITIONS • Omit both	Any of the following combinations: • MAXPARTITIONS and NUMPARTS • MAXPARTITIONS and SEGSIZE n[1] on page 90 • MAXPARTITIONS
Partition-by-range	NUMPARTS only	NUMPARTS and SEGSIZE n[1] on page 90
Segmented (non-UTS)	Not supported[2] on page 90	One of the following combinations: • SEGSIZE n[1] on page 90 • Omit MAXPARTITIONS, NUMPARTS, and SEGSIZE
Partitioned (non-UTS)	Not supported[2] on page 90	NUMPARTS and SEGSIZE 0

Notes:

1. Where n is a non-zero value. The DPSEGSZ subsystem parameter determines the default value. For more information, see DEFAULT PARTITION SEGSIZE field (DPSEGSZ subsystem parameter) (Db2 Installation and Migration).
2. FL 504 Non-UTS table spaces for base tables are deprecated. CREATE TABLESPACE statements that run at application compatibility level V12R1M504 or higher always create a partition-by-growth or partition-by-range table space, and CREATE TABLE statements that specify a non-UTS table space (including existing multi-table segmented table spaces) return an error. However, you can use a lower application compatibility level to create table spaces of the deprecated types if needed, such as for recovery situations. For instructions, see Creating non-UTS table spaces (deprecated) (Db2 Administration Guide).

Also, a CREATE TABLE statement to create a new table in a segmented (non-UTS) table space results in an error.

For more information, see Table space types and characteristics in Db2 for z/OS (Db2 Administration Guide) and CREATE TABLESPACE (Db2 SQL).

Hash-organized tables

With the release of function level 504, hash organized tables are now considered deprecated. CREATE TABLE or ALTER TABLE statement that specifies ORGANIZE BY HASH results in an error. Newer capabilities such as fast index traversal in Db2 12 offer a more effective solution than hash-organized tables. For more information, see Fast index traversal (Db2 Performance).

APAR PH02873 delivered the functional code that supports preventing the creation of the deprecated objects.

Newly supported built-in functions with IBM Db2 Analytics Accelerator

Function level 504 introduces support for the following passthrough-only built-in functions, which are passed through from Db2 for z/OS to IBM Db2 Analytics Accelerator:

- CUME_DIST
- CUME_DIST (aggregate) (Db2 SQL)
- FIRST_VALUE
- LAG
- LAST_VALUE
- LEAD
- NTH_VALUE
- NTILE
- PERCENT_RANK
- PERCENT_RANK (aggregate) (Db2 SQL)
- RATIO_TO_REPORT
- REGEXP_COUNT (Db2 SQL)
- REGEXP_INSTR (Db2 SQL)
- REGEXP_LIKE (Db2 SQL)
- REGEXP_REPLACE (Db2 SQL)
- REGEXP_SUBSTR (Db2 SQL)

Db2 for z/OS only verifies that the data types of the parameters are valid for the functions. The accelerator engine does all other function resolution processing and validation. For more information, see Accelerating queries with passthrough-only expressions.

APAR PH00224 delivered the functional code for the new support of these passthrough-only built-in functions.

New SQL syntax alternatives for special registers and NULL predicates

Function level 504 introduces new syntax alternatives in Db2 for z/OS to simplify the porting of database applications from other platforms. For example, support is introduced in Db2 for z/OS for the following alternative syntax for referencing existing special registers. For more information, see Special registers (Db2 SQL)

Table 5. New syntax alternatives for special registers in function level 504

Existing Special Register	New Syntax Alternative	Number of Keywords for New Syntax Alternative
CURRENT CLIENT_ACCTNG	CLIENT ACCTNG	2
CURRENT CLIENT_APPLNAME	CLIENT APPLNAME	2
CURRENT CLIENT_USERID	CLIENT USERID	2
CURRENT CLIENT_WRKSTNNAME	CLIENT WRKSTNNAME	2
CURRENT SERVER	CURRENT_SERVER	1 "1" on page 92
CURRENT TIME ZONE CURRENT TIMEZONE	CURRENT_TIMEZONE	1 "1" on page 92

Notes:

1. The new single-token syntax alternatives might introduce an incompatible change if existing data or applications use a column or variable with the same name as the single token.

The following table shows the new syntax alternatives for the NULL predicate that function level 504 introduces in Db2 for z/OS. For more information, see NULL predicate (Db2 SQL).

Table 6. New syntax alternatives for the NULL predicate in function level 504

Existing Predicate	New Syntax Alternatives
IS NULL	ISNULL
IS NOT NULL	NOTNULL

APAR PH02646 delivered the functional code to support the new SQL syntax alternatives.

Activation details for function level 504

Function level 504 has the following activation details:

Enabling APAR:	PH07672.
Minimum Db2 catalog level:	V12R1M503
Application compatibility control:	Applications must run at application compatibility level V12R1M504 or higher to use the following new capabilities: • All new SQL syntax described in "New SQL syntax alternatives for special registers and NULL predicates" on page 91. • All built-in functions described in "Newly supported built-in functions with IBM Db2 Analytics Accelerator" on page 91. • Preventing the creation of deprecated object types, as described in "Prevent creation of new deprecated objects" on page 89.

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 12 environment and continuous delivery in general, see Part 2, "Adopting new capabilities in Db2 12 continuous delivery," on page 261.

To activate Db2 12 function level 504, complete the following steps:

1. If Db2 12 is still at function level 100, activate function level 500 or 501 first. For more information, see Activating Db2 12 new function at migration (Db2 Installation and Migration).

2. Verify that any incompatible changes are resolved, including for any lower function levels not yet activated. See "Incompatible changes summary for function levels 501 and higher" on page 119.

3. Check that Db2 is at a sufficient code level by issuing a DISPLAY GROUP command, as described in Chapter 9, "Determining the Db2 code level, catalog level, and function level," on page 271. The DSN7100I message indicates the Db2 code level for under DB2 LVL in the member details. If DB2 LVL for any member is lower than 121504, apply the PTF for APAR PH07672 and any other required maintenance, so that all members indicate 121504.

   Tip: You can apply any PTF at any function level. It is best to run Db2 at this code level or higher for some time before you proceed with activating the function level. Db2 cannot run at any lower code level after you activate a function level, so you cannot remove any of the required PTFs after you activate a function level.

4. Tailor the CATMAINT and function level activation jobs by running the installation CLIST:

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 ===> V12R1M504. 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).

5. Run the DSNTIJIC job to take an image copy of the Db2 catalog and directory.

6. Run the DSNTIJTC job, or run the CATMAINT utility with LEVEL V12R1M504, to update the 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.

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

```
-ACTIVATE FUNCTION LEVEL (V12R1M504) TEST
```

Db2 issues message DSNU757I to indicate the results. For more information, see Chapter 10, "Testing Db2 function level activation," on page 277.

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

```
-ACTIVATE FUNCTION LEVEL (V12R1M504)
```

9. 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 15, "Controlling the Db2 application compatibility level," on page 289.

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.

Function level 504 incompatible changes

The following changes might impact your Db2 environment and applications that run at application compatibility level V12R1M504 or higher, such as the following changes:

- If your Db2 applications have unqualified references to existing user-defined functions, they might start invoking new built-in functions instead if the names and signatures match, in certain situations. For information about avoiding such situations, see Ensuring that Db2 executes the intended user-defined function (Db2 Application programming and SQL).

- If your Db2 environment contains existing columns or variables with names such as CURRENT_TIMEZONE or CURRENT_SERVER, you must modify your applications to delimit these names. For example, you can issue the following queries to identify columns and variables with these names. For columns, issue:

```
SELECT * FROM SYSIBM.SYSCOLUMNS
  WHERE NAME IN('CURRENT_SERVER','CURRENT_TIMEZONE');
```

For variables, issue:

```
SELECT * FROM SYSIBM.SYSVARIABLES
WHERE NAME IN('CURRENT_SERVER','CURRENT_TIMEZONE');
```

Based on the result, you can then check for dependencies, for example by querying the SYSIBM.SYSPACKDEP catalog table, to identify any static applications that depend on the nondelimited forms of these names.

- CREATE TABLESPACE statements that previously created non-UTS segmented or partitioned table spaces, now always create partition-by-growth or partition-by-range table spaces.

  Tip: Although it is best to use the non-deprecated types for new objects, you can still use a lower application compatibility level to create the deprecated table spaces or higher if necessary, such as for recovery situations. For instructions, see Creating non-UTS table spaces (deprecated) (Db2 Administration Guide).

- Errors can be returned for SQL statements that previously succeeded, as side effects of the tables now being created in universal table spaces instead of non-UTS types. For example:

  – CREATE AUX TABLE statements for LOB columns can start failing with SQLCODE -769 because the PART clause is required when the base table resides in a PBG table space.

  – DROP TABLE statements can start failing with SQLCODE -669 because the DROP TABLE statement is not supported for tables in PBG table spaces. The DROP TABLESPACE statement must be used instead. (Function level 506 resolves this situation. See "Support for implicitly dropping explicitly created UTS and LOB table spaces" on page 82.)

  – CREATE TABLE statements can start failing with SQLCODE -646 if they attempt to create more than one table in a table space. Each universal table space contains only single table.

- The following SQL statements now return errors:

  – CREATE SYNONYM statements

  – CREATE TABLE statements that specify existing non-UTS table spaces

  – CREATE TABLE and ALTER TABLE statements that specify ORGANIZE BY HASH

  Action to take: Update your applications to use the non-deprecated object types.

Function level 503 (PH00506 - October 2018)

Function level 503 (V12R1M503) introduces support for replication of system-period temporal tables and generated expression columns, a change to temporal auditing support for temporal data, and a new console message that indicates when catalog levels or function levels change.

Contents

Finding function level 503 changes: To find the new and changed content for this function level, try searching for "FL 503" from any page. You'll see a list of new and changed topics for function level 503.

Throughout the Db2 12 information, when you see the link FL 503, the adjacent content was changed for function level 503, and you can click the link to see the page that you are currently reading.

About Db2 AI for z/OS support:

Function level 503 originally introduced support for IBM Db2 AI for z/OS, which leverages machine learning technology to empower the Db2 for z/OS optimizer to determine the best-performing query access paths, based on your workload characteristics.

However, with APAR PH05323 and IBM Db2 AI for z/OS Version 1.1.0.1 or later, Db2 12 supports IBM Db2 AI for z/OS at function level 500 or higher.

For more information see IBM Db2 AI for z/OS documentation.

Enablement for replication of system-period temporal tables and generated expression columns

Function level 503 provides support needed by replication programs to replicate Db2 for z/OS data in system-period temporal tables and in tables that contain generated expression columns.

A new built-in global variable, SYSIBMADM.REPLICATION_OVERRIDE, provides support needed to replicate system-period temporal tables or in tables that contain generated expression columns. The global variable is intended for use on the apply side of the replication process by products that enable replication of Db2 data. Applications that set the SYSIBMADM.REPLICATION_OVERRIDE global variable must be bound using APPLCOMPAT V12R1M503.

See REPLICATION_OVERRIDE (Db2 SQL) for more information.

Console message for catalog level or function level change

Function level 503 introduces console message DSNG014I. It is issued to the console whenever the function level or catalog level of a Db2 subsystem or data sharing group changes. See DSNG014I (Db2 Messages).

Temporal auditing

When a system-period temporal table that contains one or more rows of data is altered to add a DATA CHANGE OPERATION generated expression column (also called an *auditing column*), Db2 assigns null values to the new column in the existing rows. If the table was also defined with the ON DELETE ADD EXTRA ROW attribute, prior to function level 503, temporal queries that were run on these types of tables did not return any rows that contained a null value that corresponds to the DATA CHANGE OPERATION column. As a result, the data that was returned did not include the historical data that was expected.

With function level 503, rows that contain null values in the history table column that corresponds to the DATA CHANGE OPERATION column will now be considered part of the intermediate result set for a system-period temporal query. However, rows that were added to a history table for the ON DELETE ADD EXTRA ROW attribute will still be excluded.

For information about the auditing capabilities of Db2, see Scenario for tracking auditing information (Db2 Administration Guide).

APAR PI95480 delivered functional code to support the new temporal auditing capability.

Activation details for function level 503

Function level 503 has the following activation details:

Enabling APAR:	PH00506
Minimum Db2 catalog level:	V12R1M503
Catalog level V12R1M503 changes:	The REPLICATION_OVERRIDE built-in global variable is added. For a summary of changes in Db2 12, see "Catalog changes in Db2 12" on page 179.

Application compatibility control:	Applications must run at application compatibility level V12R1M503 or higher to use the following new capabilities:
	• The SYSIBMADM.REPLICATION_OVERRIDE built-in global variable for replication of system-period temporal tables and generated expression columns.
	• The temporal query result change for system-period temporal tables defined with the ON DELETE ADD EXTRA ROW attribute that also contain a DATA CHANGE OPERATION column.

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 12 environment and continuous delivery in general, see Part 2, "Adopting new capabilities in Db2 12 continuous delivery," on page 261.

To activate Db2 12 function level 503, complete the following steps:

1. If Db2 12 is still at function level 100, activate function level 500 or 501 first. For more information, see Activating Db2 12 new function at migration (Db2 Installation and Migration).

2. Verify that any incompatible changes are resolved, including for any lower function levels not yet activated. See "Incompatible changes summary for function levels 501 and higher" on page 119.

3. Check that Db2 is at a sufficient code level by issuing a DISPLAY GROUP command, as described in Chapter 9, "Determining the Db2 code level, catalog level, and function level," on page 271. The DSN7100I message indicates the Db2 code level for under DB2 LVL in the member details. If DB2 LVL for any member is lower than 121503, apply the PTF for APAR PH00506 and any other required maintenance, so that all members indicate 121503.

   Tip: You can apply any PTF at any function level. It is best to run Db2 at this code level or higher for some time before you proceed with activating the function level. Db2 cannot run at any lower code level after you activate a function level, so you cannot remove any of the required PTFs after you activate a function level.

4. Tailor the CATMAINT and function level activation jobs by running the installation CLIST:

   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 ===> V12R1M503. 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).

5. Run the DSNTIJIC job to take an image copy of the Db2 catalog and directory.

6. Run the DSNTIJTC job, or run the CATMAINT utility with LEVEL V12R1M503, 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.

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

```
-ACTIVATE FUNCTION LEVEL (V12R1M503) TEST
```

Db2 issues message DSNU757I to indicate the results. For more information, see Chapter 10, "Testing Db2 function level activation," on page 277.

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

```
-ACTIVATE FUNCTION LEVEL (V12R1M503)
```

9. 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 15, "Controlling the Db2 application compatibility level," on page 289.

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.

Function level 503 incompatible changes

Function level 503 introduces the following incompatible changes.

Result change for system-period temporal tables defined with ON DELETE ADD EXTRA ROW

A query has incompatible behavior at application compatibility level V12R1M503 if all of the following conditions are met:

- It references a system-period temporal table (STT) with FOR SYSTEM_TIME FROM value1 TO value2 (or BETWEEN value1 AND value2) period specification. The STT can be directly referenced or indirectly referenced through views.
- The STT is defined with the ON DELETE ADD EXTRA ROW clause.
- The STT is defined with the DATA CHANGE OPERATION column.
- During migration to use the temporal auditing feature, the altered table contains existing rows.
- The DATA CHANGE OPERATION column is populated with the default NULL value for the existing rows. When the query is executed, the historical table contains records with a NULL value in DATA CHANGE OPERATION column.

Action required: Start a trace for IFCID 0376. In the trace output, function code 1215031 or 1202 identifies the affected queries.

The SET_MAINT_MODE_RECORD_NO_TEMPORALHISTORY stored procedure is not supported when APPLCOMPAT is V12R1M503 or higher

Users will not be able to invoke the SYSPROC.SET_MAINT_MODE_RECORD_NO_TEMPORALHISTORY stored procedure when running with APPLCOMPAT=V12R1M503 or higher. Existing applications that invoke the SYSPROC.SET_MAINT_MODE_RECORD_NO_TEMPORALHISTORY stored procedure will fail starting with APPLCOMPAT=V12R1M503 or higher.

Action required: Start a trace for IFCID 0376. In the trace output, function code 1215032 or 1203 identifies applications that use the SYSPROC.SET_MAINT_MODE_RECORD_NO_TEMPORALHISTORY stored procedure. Modify such applications to set value of the SYSIBMADM.REPLICATION_OVERRIDE built-in global variable to 'Y' instead. For more information, see SYSIBMADM.REPLICATION_OVERRIDE built-in global variable.

For all incompatible changes in Db2 12, see "Incompatible changes in Db2 12" on page 119.

Function level 502 (PI95511 - May 2018)

Function level 502 (V12R1M502) introduces management of key labels for z/OS DFSMS data set encryption and explicit casting of numeric values to GRAPHIC or VARGRAPHIC.

Contents

Finding function level 502 changes: To find the new and changed content for this function level, try searching for "FL 502" from any page. You'll see a list of new and changed topics for function level 502.

Throughout the Db2 12 information, when you see the link FL 502 , the adjacent content was changed for function level 502, and you can click the link to see the page that you are currently reading.

Key label management for z/OS DFSMS data set encryption

Function level 502 introduces management of key labels for z/OS DFSMS data set encryption to transparently encrypt Db2 data sets. DFSMS can be used to encrypt various types of Db2 data sets including Db2-managed table space and index space data sets, data sets that are used by Db2 utilities, and sequential input and output data sets.

> Encrypting log, catalog, and directory data sets with z/OS DFSMS data set encryption (Managing Security)
> Encrypting all data sets using a storage group with z/OS DFSMS data set encryption (Managing Security)
> Using z/OS DFSMS data set encryption to encrypt the data sets associated with a particular table (Managing Security)
> Changing Db2 data sets from encrypted to unencrypted (Managing Security)
> Encrypted FlashCopy image copies, copies made with DFSMSdss concurrent copy, and system-level backups (Managing Security)
> Sample procedure for setting up z/OS DFSMS data set encryption (Managing Security)

APAR PI90288 delivered the functional code to support key label management for z/OS DFSMS data set encryption.

Explicit casting of numeric values to GRAPHIC or VARGRAPHIC

Function level 502 introduces casting an explicit numeric value to a graphic string value. All of the numeric data types (SMALLINT, INTEGER, BIGINT, DECIMAL, REAL, DOUBLE, FLOAT, and DECFLOAT) are supported. You can use the GRAPHIC or VARGRAPHIC built-in functions and the CAST specification for the explicit cast.

When the GRAPHIC or VARGRAPHIC functions are used to explicitly cast a decimal value to graphic, an optional decimal separator character can be specified as a single-byte character constant for the second argument. The extended support for the GRAPHIC and VARGRAPHIC built-in functions to allow numeric input data is similar to the existing support for numeric input to the CHAR and VARCHAR built-in functions.

Whether the CAST specification or the GRAPHIC or VARGAPHIC functions are used, the result is Unicode (UTF-16), and the context must support Unicode data.

> CAST specification (Db2 SQL)

GRAPHIC (Db2 SQL)
VARGRAPHIC (Db2 SQL)
Casting between data types (Db2 SQL)

APAR PI88790 delivered the functional code to support explicit casting of numeric values to GRAPHIC or VARGRAPHIC.

Db2 12 migration process changes

The Db2 12 migration process is modified to facilitate activation of function level 502 from function level 100, or as part of migration from Db2 11 (with z/OSMF only). Function level 502 requires catalog level 502, and tailoring the catalog for level 502 requires function level 500 or 501. To help manage these requirements, function level 502 introduces the following changes to the Db2 12 migration process:

- Job DSNTIJA1 is added for the intermediate activation of function level 501. DSNTIJA1 consists of a single job step that activates function level 501. It is generated by the installation CLIST only in the following circumstances:

  - In non-z/OSMF ACTIVATE mode, when activating function level 502 or higher from function level 100. In non-z/OSMF MIGRATE mode, DSNTIJA1 is not generated because activating function level 500 or higher is not built into that process. Function level 500 or higher is always activated later, by running the CLIST in ACTIVATE mode.

  - In z/OSMF MIGRATE mode, when migrating a Db2 subsystem or the first member only of a data sharing group to function level 502. In z/OSMF ACTIVATE mode, DSNTIJA1 is not generated because activating function level 500 or higher is built into the z/OSMF migration process.

- In ACTIVATE mode, the CLIST is enhanced to compare the current Db2 function level to the specified target function level, which are specified on panel DSNTIP00, and generates the required job set.

- On panel DSNTIP00:

  - A new field CURRENT FUNCTION LEVEL is added. It is only used in ACTIVATE mode. It is initially set to the input DSNTID*xx* FUNCLVL parameter and can be changed. In other modes, this field is always blank and protected.

  - The existing DB2 FUNCTION LEVEL field is renamed to TARGET FUNCTION LEVEL.

- On panel DSNTIPMN, a new step is added to activate function level 501 (DSNTIJA1) before the step for tailoring the Db2 catalog (DSNTIJTC).

- In the z/OSMF workflow DSNTIWMN, a new step is added to activate function level 501 (DSNTIJA1) before the step for tailoring the catalog (DSNTIJTC).

Activation details for function level 502

Function level 502 has the following activation details:

Enabling APAR:	PI95511
Minimum Db2 catalog level:	V12R1M502
Catalog level V12R1M502 changes:	The KEYLABEL column is added in the following catalog tables: • SYSIBM.SYSINDEXES • SYSIBM.SYSSTOGROUP • SYSIBM.SYSTABLESPACE • SYSIBM.SYSTABLES **Restriction:** Before you can run CATMAINT to tailor the Db2 catalog for function level 502 or higher, you must first activate function level 500 or 501. That is, the DISPLAY GROUP command output must indicate `HIGHEST ACTIVATED FUNCTION LEVEL (V12R1M500)`

	or higher. This restriction prevents tailoring the Db2 catalog for function levels higher than 500 while fallback to Db2 11 remains possible. However, later activating a lower function level such as function level 100* does not restrict the CATMAINT operation. For a summary of changes in Db2 12, see "Catalog changes in Db2 12" on page 179.
Application compatibility control:	Applications must run at application compatibility level V12R1M502 or higher to use the following new capabilities: • CREATE or ALTER statements that specify the KEYLABEL option. However, applications that already reference encrypted data can continue to run at their current application compatibility levels. • Explicit casting of numeric data types to fixed or variable length graphic strings, with the GRAPHIC or VARGRAPHIC built-in functions or the CAST specification.

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 12 environment and continuous delivery in general, see Part 2, "Adopting new capabilities in Db2 12 continuous delivery," on page 261.

To activate Db2 12 function level 502, complete the following steps:

1. If Db2 12 is still at function level 100, activate function level 500 or 501 first. For more information, see Activating Db2 12 new function at migration (Db2 Installation and Migration).

2. Verify that any incompatible changes are resolved, including for any lower function levels not yet activated. See "Incompatible changes summary for function levels 501 and higher" on page 119.

3. Check that Db2 is at a sufficient code level by issuing a DISPLAY GROUP command, as described in Chapter 9, "Determining the Db2 code level, catalog level, and function level," on page 271. The DSN7100I message indicates the Db2 code level for under DB2 LVL in the member details. If DB2 LVL for any member is lower than 121502, apply the PTF for APAR PI95511 and any other required maintenance, so that all members indicate 121502.

   Tip: You can apply any PTF at any function level. It is best to run Db2 at this code level or higher for some time before you proceed with activating the function level. Db2 cannot run at any lower code level after you activate a function level, so you cannot remove any of the required PTFs after you activate a function level.

4. Tailor the CATMAINT and function level activation jobs by running the installation CLIST:

   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 ===> V12R1M502. 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).

5. Run the DSNTIJIC job to take an image copy of the Db2 catalog and directory.

6. Run the DSNTIJTC job, or run the CATMAINT utility with LEVEL V12R1M502, to update the 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.

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

```
-ACTIVATE FUNCTION LEVEL (V12R1M502) TEST
```

Db2 issues message DSNU757I to indicate the results. For more information, see Chapter 10, "Testing Db2 function level activation," on page 277.

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

```
-ACTIVATE FUNCTION LEVEL (V12R1M502)
```

9. 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 15, "Controlling the Db2 application compatibility level," on page 289.

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.

Function level 502 incompatible changes

If your environment contains existing user-defined functions named GRAPHIC or VARGRAPHIC, applications with unqualified references to the existing user-defined functions might start invoking the built-in function instead of the user-defined function in certain situations. For information about avoiding such situations, see Ensuring that Db2 executes the intended user-defined function (Db2 Application programming and SQL).

For all incompatible changes in Db2 12, see "Incompatible changes in Db2 12" on page 119

Function level 501 (PI70535 - May 2017)

Function level 501 (V12R1M501) introduces the LISTAGG built-in function, which produces a list of all values in a group.

Contents

The LISTAGG built-in function

The LISTAGG built-in function produces a list of all values in a group. An optional separator argument can delimit items in the result list. For example, specifying a comma as the separator produces a comma-separated list. An optional ordering can also be specified for the items within the group.

Example

Produce an alphabetical list of comma-separated names, grouped by department, from the sample employee table.

```
SELECT WORKDEPT,
  LISTAGG(LASTNAME, ', ') WITHIN GROUP(ORDER BY LASTNAME)
  AS EMPLOYEES
  FROM EMP
  GROUP BY WORKDEPT;
```

The following result is returned.

```
---------+---------+---------+---------+---------+---------+---------+---------+---------+-------
WORKDEPT  EMPLOYEES
---------+---------+---------+---------+---------+---------+---------+---------+---------+-------
A00      HAAS, HEMMINGER, LUCCHESI, O'CONNELL, ORLANDO
B01      THOMPSON
C01      KWAN, NATZ, NICHOLLS, QUINTANA
D11      ADAMSON, BROWN, JOHN, JONES, LUTZ, PIANKA, SCOUTTEN, STERN, WALKER, YAMAMOTO, YOSHIMURA
D21      JEFFERSON, JOHNSON, MARINO, MONTEVERDE, PEREZ, PULASKI, SMITH
E01      GEYER
E11      HENDERSON, PARKER, SCHNEIDER, SCHWARTZ, SETRIGHT, SMITH, SPRINGER
E21      ALONZO, GOUNOT, LEE, MEHTA, SPENSER, WONG
```

APAR PI73929 delivered functional code to support the LISTAGG built-in function.

Function level 501 activation details

Function level 501 has the following activation details:

Enabling APAR:	PI70535
Minimum required catalog level:	V12R1M500
Application compatibility control:	Use of the LISTAGG built-in function level requires application compatibility level V12R1M501 or higher.
	IBM data server clients and drivers that use the capabilities in function level 501 or are bound with application compatibility level V12R1M501 or higher, require extra program preparation steps. For instructions, see Setting application compatibility levels for data server clients and drivers (Db2 Application programming and SQL).

How to activate function level 501

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 12 environment and continuous delivery in general, see Part 2, "Adopting new capabilities in Db2 12 continuous delivery," on page 261.

To activate Db2 12 function level 501, complete the following steps:

1. For a new Db2 12 installation, skip Step 3. Otherwise, verify that fallback to Db2 11 and coexistence in data sharing are no longer required in your Db2 12 environment.

2. If Db2 was migrated from Db2 11, and function level 500 is not yet activated, follow the steps in Activating Db2 12 new function at migration (Db2 Installation and Migration).

3. Check that Db2 is at a sufficient code level by issuing a DISPLAY GROUP command, as described in Chapter 9, "Determining the Db2 code level, catalog level, and function level," on page 271. The DSN7100I message indicates the Db2 code level for under DB2 LVL in the member details. If DB2 LVL for any member is lower than 121501, apply the PTF for APAR PI70535 and any other required maintenance, so that all members indicate 121501.

Tip: You can apply any PTF at any function level. It is best to run Db2 at this code level or higher for some time before you proceed with activating the function level. Db2 cannot run at any lower code level after you activate a function level, so you cannot remove any of the required PTFs after you activate a function level.

4. Tailor the CATMAINT and function level activation jobs by running the installation CLIST:

   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 ===> V12R1M501. 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).

5. Run the DSNTIJIC job to take an image copy of the Db2 catalog and directory.

6. Run the DSNTIJTC job, or run the CATMAINT utility with LEVEL V12R1M501, to update the 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.

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

```
-ACTIVATE FUNCTION LEVEL (V12R1M501) TEST
```

   Db2 issues message DSNU757I to indicate the results. For more information, see Chapter 10, "Testing Db2 function level activation," on page 277.

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

```
-ACTIVATE FUNCTION LEVEL (V12R1M501)
```

9. 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 15, "Controlling the Db2 application compatibility level," on page 289.

   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.

Function level 501 incompatible changes

Function level 501 introduces no incompatible changes.

For all incompatible changes in Db2 12, see "Incompatible changes in Db2 12" on page 119

Related reference
LISTAGG (Db2 SQL)

Function level 500 (installation or migration - October 2016)

Function level 500 (V12R1M500) represents the first opportunity for applications to take advantage of most new capabilities in the Db2 12 initial release, including new SQL capabilities and many new subsystem parameter settings. Activation of function level 500 prevents any future fallback to Db2 11. Function level 500 is comparable in many respects to new-function mode in previous releases.

In Db2 12, function level 500 is activated by default in new installations. However, you can activate a higher function level during the installation process.

Important: Before you activate function level 500 or higher for the first time, read Activating Db2 12 new function at migration (Db2 Installation and Migration).

Important: Do not issue the ACTIVATE command or run job DSNTIJAF for activation of Db2 12 until you are certain that the subsystem or data sharing group can proceed on Db2 12, without the possibility of falling back to or coexistence with Db2 11. In data sharing, the ACTIVATE command has group scope. Fallback and coexistence become impossible with the successful activation of function level 500 or higher.

Important: Apply the for fallback SPE (APAR PI33871) and stop and restart Db2 11 for every subsystem or data sharing member that you plan to migrate to Db2 12. For data sharing, every member must be started in Db2 11 after the fallback SPE is applied. Inactive members that never started with the fallback SPE applied in Db2 11 cannot start in Db2 12 or Db2 11 after migration to Db2 12 and activation of function level 500 on any other member. See Required maintenance for Db2 12 installation or migration (Db2 Installation and Migration).

Contents

Changes in Function level 500

To read more about many of new capabilities in Db2 12, see the following topics:

The following sections summarize changes that function level 500 introduces in Db2 12.

Subsystem parameter changes in function level 500

The following changes take effect when you activate function level 500 in Db2 12.

Subsystem parameter	Change introduced	Incompatible change?
AUTH_COMPATIBILITY in macro DSN6SPRM (Db2 Installation and Migration)	New subsystem parameter. ***** This new subsystem parameter is not an incompatible change. However, its purpose is to provide time to resolve specific authorization-related incompatible changes, and the default value specifies that Db2 12 uses the new authorization behavior. See "Changes to the authorization	No *

Subsystem parameter	Change introduced	Incompatible change?
	check of the UNLOAD utility" in "Security release incompatibilities" on page 140.	
CACHE DYN STABILITY field (CACHEDYN_STABILIZATION subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
COMPRESS DB2 DIR LOBS field (COMPRESS_DIRLOB subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
DDL MATERIALIZATION field (DDL_MATERIALIZATION subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
DEFAULT INSERT ALGORITHM field (DEFAULT_INSERT_ALGORITHM subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
PAGE SET PAGE NUMBERING field (PAGESET_PAGENUM subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
REORG IC LIMIT DASD field (REORG_IC_LIMIT_DASD subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
REORG IC LIMIT TAPE field (REORG_IC_LIMIT_TAPE subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
UTILS BLOCK FOR CDC field (UTILS_BLOCK_FOR_CDC subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
REORG INDEX NOSYSUT1 (REORG_INDEX_NOSYSUT1 subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	
SUBSTR COMPATIBILITY field (SUBSTR_COMPATIBILITY subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No

Command changes in function level 500

The following changes to commands take effect when you activate function level 500 in Db2 12.

Command	Change introduced	Incompatible change?
-ACCESS DATABASE (Db2) (Db2 Commands)	If the externalization of real-time statistics (RTS) fails, Db2 now issues message DSNT538I.	Yes
-ALTER BUFFERPOOL (Db2) (Db2 Commands)	The meaning of the existing PGSTEAL(NONE) option is changed. As in previous versions, Db2 pre-loads the data for assigned objects when they are opened and keeps the data resident until the objects are closed. However, in Db2 12, Db2 creates an overflow area within the VPSIZE of the buffer pool. If all open assigned objects do not fit into the main part of the buffer pool, pages are allocated to the overflow area. Page stealing is possible in the overflow area. After pages are allocated in the overflow area, you must close the object to move the pages back to the overflow area.	No
-ALTER GROUPBUFFERPOOL (Db2) (Db2 Commands)	The maximum value that can be specified for the RATIO keyword is now increased to 1024.	No
BIND PACKAGE subcommand (DSN) (Db2 Commands)	The following new option is added: CONCENTRATESTMT.	No
BIND QUERY (DSN) (Db2 Commands)	Support is added for processing the following SQL statement types: • INSERT • UPDATE WHERE CURRENT OF • DELETE WHERE CURRENT OF	No
-DISPLAY DYNQUERYCAPTURE (Db2) (Db2 Commands)	New command.	No
-DISPLAY GROUP (Db2) (Db2 Commands)	The DSN7100I message output is modified to indicate information about the code, catalog, and function levels of the Db2 subsystem or data sharing group.	No
-DISPLAY ML (Db2) (Db2 Commands)	New command. [1] on page 108	No.
-DISPLAY STATS (Db2) (Db2 Commands)	New command. New options: • INDEXTRAVERSECOUNT is used to display the index traverse counts for a specific index, or for a specified number of indexes with the highest traverse counts, in descending order. Output is returned by message DSNT830I (Db2 Messages). • LOGREADERTASKS and LOGREADERTASKS SCOPE(GROUP) options are used to display statistics about any log reading tasks that are	No

Command	Change introduced	Incompatible change?
	currently running. Output is returned by message DSNT788I.	
-DISPLAY UTILITY (Db2) (Db2 Commands)	The output now includes message DSNU285I.	Yes
FREE STABILIZED DYNAMIC QUERY (DSN) (Db2 Commands)	New command.	No
FREE PACKAGE (DSN) (Db2 Commands)	If you specify INACTIVE, PREVIOUS, ORIGINAL for the PLANMGMTSCOPE option, FREE PACKAGE processing can now complete while applications that use the package are running. New values are added for an existing option: PLANMGMTSCOPE(PREVIOUS|ORIGINAL). A new option is added: INVALIDONLY A new value is added to an existing option: PLANMGMTSCOPE(PHASEOUT).	No
FREE SERVICE (DSN) (Db2 Commands)	New command. SERVICE now supports specifying *location-name* and *version-id*.	No
-MODIFY DDF (Db2) (Db2 Commands)	After the PTF for APAR PH08188 is applied, PORT and SECPORT can now have the name value. A new option is added: SESSIDLE.	No
REBIND PACKAGE subcommand (DSN) (Db2 Commands)	This command can be used to rebind a package for an advanced trigger. The following new options are added: • APREUSESOURCE • CONCENTRATESTMT If the SWTICH option is specified with an invalid package, the command now fails.	Yes
REBIND TRIGGER PACKAGE (DSN) (Db2 Commands)	This command can be used to rebind a package for a basic trigger, including any trigger that was created prior to the activation of function level 500 or higher. A new option is added: APREUSESOURCE. If the SWTICH option is specified with an invalid package, the command now fails.	Yes
-START DB2 (Db2) (Db2 Commands)	You cannot start Db2 at any code level that is lower than the catalog level or the highest activated function level.	No
-START PROFILE (Db2) (Db2 Commands)	If the PROFILE_AUTOSTART subsystem parameter is set to YES, this command is now issued automatically when Db2 starts.	No

Command	Change introduced	Incompatible change?
-START ML (Db2) (Db2 Commands)	New command. [1] on page 108 New options added: *, SQLOPT, and SA_DCC.	No
-START RLIMIT (Db2) (Db2 Commands)	Certain previously supported resource limit table formats are deprecated or unsupported in Db2 12. Resource limits are not activated for tables with unsupported formats.	No
-STOP ML (Db2) (Db2 Commands)	New command. [1] on page 108 New options added: *, SQLOPT, and SA_DCC.	No
-TERM UTILITY (Db2) (Db2 Commands)	The TERM UTILITY command can be used for the MODIFY RECOVERY utility.	No
-START CDDS (Db2) (Db2 Commands)	New command.	No
-STOP CDDS (Db2) (Db2 Commands)	New command.	No
-START DYNQUERYCAPTURE (Db2) (Db2 Commands)	New command.	No
-STOP DYNQUERYCAPTURE (Db2) (Db2 Commands)	New command.	No

Notes:

1. With IBM Db2 AI for z/OS Version 1.1.0.1 or later and APAR PH05323.

SQL statement changes in function level 500

The following SQL changes take effect in Db2 12 for applications that run at application compatibility V12R1M500 or higher.

SQL element	Change introduced	Incompatible change?
ALTER FUNCTION (compiled SQL scalar)	New clause: CONCENTRATE STATEMENTS.	No
ALTER INDEX	A new clauses are added: DSSIZE. The following clauses are changed: • ADD COLUMN *column-name* (can now specify DECFLOAT columns) • COMPRESS Altering to use index compression for indexes in universal table spaces is now a pending change that places the index in advisory REORG-pending (AREOR) status	Yes
ALTER PROCEDURE (SQL native)	New clause: CONCENTRATE STATEMENTS.	No
ALTER TABLE	The following new clauses are added:	No

SQL element	Change introduced	Incompatible change?
	• CCSID • EXCLUSIVE and INCLUSIVE clauses of the PERIOD BUSINESS_TIME clause • PERIOD BUSINESS_TIME clause for constraints The following clauses are changed: • ADD PERIOD FOR • ADD PARTITION	
ALTER TABLESPACE	The following new clauses are added: • PAGENUM • INSERT ALGORITHM The following clauses are changed: • COMPRESS • DSSIZE	No
ALTER TRIGGER (advanced) (Db2 SQL)	New statement.	No
ALTER TRIGGER (basic)	Equivalent to ALTER TRIGGER in prior releases.	No
COMMENT (Db2 SQL)	Changed clause: TRIGGER *trigger-name* VERSION *trigger-version-id*.	No
CREATE FUNCTION (compiled SQL scalar)	The following new clauses are added: • CONCENTRATE STATEMENTS • WRAPPED	No
CREATE FUNCTION (inline SQL scalar)	New clause: WRAPPED.	No
CREATE FUNCTION (SQL table)	New clause: WRAPPED.	No
CREATE INDEX	New clause: DSSIZE. CREATE INDEX statements that omit the USING clause at the table space or index level now fail with SQLCODE -204, if the storage group specified when the containing database was created does not exist.	Yes
CREATE PROCEDURE (SQL native)	The following new clauses are added: • CONCENTRATE STATEMENTS • WRAPPED	No
CREATE TABLE	The following new clauses are added: • CCSID on a CHAR, GRAPHIC, CLOB, or DBCLOB column • EXCLUSIVE and INCLUSIVE clauses of the PERIOD BUSINESS_TIME clause • PAGENUM • PERIOD BUSINESS_TIME for constraints	No

SQL element	Change introduced	Incompatible change?
	The following clauses are changed: • DSSIZE • PERIOD FOR	
CREATE TABLESPACE	The following new clauses are added: • PAGENUM • INSERT ALGORITHM The following clauses are changed: • COMPRESS • DSSIZE • NUMPARTS CREATE TABLESPACE statements that omit the USING clause at the table space or index level now fail with SQLCODE -204, if the storage group specified when the containing database was created does not exist.	Yes
CREATE TRIGGER (basic)	New clause: WRAPPED.	No
CREATE TRIGGER (advanced)	New statement.	No
CREATE VARIABLE	Changed clause: *data-type*.	No
DELETE	The following new clauses are added: • BETWEEN *value-1* AND *value-2* clause of the *period-clause* • FETCH FIRST *n* ROWS ONLY	No
EXECUTE	Changed clause: USING.	No
EXECUTE IMMEDIATE (Db2 SQL)	Changed clause: *variable*.	No
EXPLAIN	New clause: STABILIZED DYNAMIC QUERY STMTID. When Db2 processes the SQL statement EXPLAIN PACKAGE or EXPLAIN STABILIZED DYNAMIC QUERY, the HINT_USED column in the PLAN_TABLE is populated with EXPLAIN PACKAGE: *copy*.	Yes
FETCH	New clause: *target-variable*.	No
fullselect (Db2 SQL)	New clause: offset-clause (Db2 SQL). The following clauses are changed: • fetch-clause (Db2 SQL) • order-by-clause (Db2 SQL)	No
GRANT (table or view privileges) (Db2 SQL)	New clause: UNLOAD.	No
GRANT (system privileges) (Db2 SQL)	New clause: BINDAGENT.	No

SQL element	Change introduced	Incompatible change?
MERGE	The MERGE statement now includes the delete operation as part of the merge process. The following new clauses are added: • *signal-statement* • ELSE IGNORE The following clauses are changed: • AS *correlation-name* • *assignment-clause* • WHEN *matching-condition* • THEN *modification-operation*	No
OPEN	Changed clause: USING.	No
PREPARE (Db2 SQL)	New clause: offset-clause (Db2 SQL). Changed clause: fetch-clause (Db2 SQL).	No
SELECT INTO (Db2 SQL)	New clause: offset-clause (Db2 SQL) The following clauses are changed: • INTO • fetch-clause (Db2 SQL)	No
SET assignment-statement (Db2 SQL)	Changed clause: DEFAULT.	No
subselect (Db2 SQL)	New clause: offset-clause (Db2 SQL) The following clauses are changed: • *correlation-clause* of collection-derived-table (Db2 SQL) in a table-reference of a FROM clause • fetch-clause (Db2 SQL) • A user-defined function that is defined with MODIFIES SQL DATA can be invoked.	No
TRANSFER OWNERSHIP (Db2 SQL)	New statement.	No
UPDATE (Db2 SQL)	New clause: BETWEEN *value-1* AND *value-2* clause of the *period-clause*.	No
VALUES INTO	The following clauses are changed: • Assignment clause source • INTO	No
compound-statement (Db2 SQL) for SQL routines	New clause: ATOMIC. Changed clause: DEFAULT or CONSTANT.	No

For more information about these changes, see "Function level 500 (installation or migration - October 2016)" on page 104.

Utility changes in function level 500

The following changes take effect when you activate function level 500 in Db2 12.

Utility	Change introduced	Incompatible change?
MODIFY RECOVERY	The following new options are added: • DELETEDS • FLASHCOPY ONLY • NOCOPYPEND When MODIFY RECOVERY is run at the table space level, a SYSIBM.SYSCOPY row with ICTYPE='M' and STYPE='R' is inserted for each partition of the table space and any partitioned indexes with the COPY YES attribute that the MODIFY utility processes. This differs from previous versions where only one SYSCOPY record was inserted for the entire table space, regardless of partitions.	No
RECOVER	FROM	No
REORG TABLESPACE	The following new options are added: • DROP_PART NO • DROP_PART YES • ICLIMIT_DASD • ICLIMIT_TAPE	No
TEMPLATE	The BLKSZLIM option is added.	No
UNLOAD (Db2 Utilities)	The authorization checking is changed, so that Db2 no longer checks for the SELECT privilege on the tables being unloaded.	Yes

Activation details for function level 500

Catalog level V12R1M500
> Catalog level V12R1M500 is the result when you install a new Db2 12 subsystem, or when you tailor the catalog during migration to Db2 12, as described in Migration step 19: Tailor Db2 12 catalog: DSNTIJTC (Db2 Installation and Migration).
>
> For a summary of changes in Db2 12, see "Catalog changes in Db2 12" on page 179.

Application compatibility level V12R1M500
> Most new SQL capabilities in function level 500 become available only to applications that use the equivalent application compatibility level (V12R1M500) or higher. For a list of such SQL capabilities, see "SQL changes in Db2 12" on page 152.
>
> For more information about application compatibility levels, see Chapter 15, "Controlling the Db2 application compatibility level," on page 289.

How to activate function level 500

To learn more about how to activate and control the adoption of new capabilities available for use in your Db2 12 environment and continuous delivery in general, see Part 2, "Adopting new capabilities in Db2 12 continuous delivery," on page 261.

In a new installation of Db2 12, function level 500 is already activated. For migration from Db2 11, complete the following steps to activation function level 500:

1. Verify that all incompatible changes are resolved. See Make adjustments for incompatible changes in Db2 12 (Db2 Installation and Migration).
2. Verify that fallback to Db2 11 and coexistence in data sharing are no longer required in your Db2 12 environment.
3. Follow the steps in Activating Db2 12 new function at migration (Db2 Installation and Migration).

Function level 500 incompatible changes

For information about incompatible changes to resolve before activating function level 500, see Make adjustments for incompatible changes in Db2 12 (Db2 Installation and Migration).

Related tasks

Activating Db2 12 function levels
You control the activation and adoption of new features in Db2 12 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.

Activating Db2 12 new function at migration (Db2 Installation and Migration)

Related reference

-ACTIVATE (Db2) (Db2 Commands)

Function level 100 (migration)

Db2 12 starts at function level 100 during migration to Db2 12. In function level 100, fallback to Db2 11 and coexistence with Db2 11 in data sharing remain possible, and most new capabilities in Db2 12 remain deactivated. Function level 100 is comparable in many respects to conversion mode in previous Db2 releases.

In function level 100 (V12R1M100), Db2 runs on Db2 12 code, and virtual storage and many optimization enhancements in Db2 12 become available. However, most new function and new SQL capabilities remain disabled until you activate the function level that introduces them.

The following types of Db2 12 enhancements are introduced in function level 100:

Virtual storage enhancements
Activation of function level 100 introduces all virtual storage enhancements in the initial Db2 12 release.

Optimization enhancements
Activation of function level 100 introduces all optimization enhancements in the initial Db2 12 release. A full prepare is required before any SQL statement can benefit from optimization enhancements. When 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

Subsystem parameters
Some new subsystem parameters and changes to existing take effect only after you activate function level 500. However, the following new and changed subsystem parameters take effect in function level 100:

Subsystem parameter	Change introduced	Incompatible change?
AUTO BIND field (ABIND subsystem parameter) (Db2 Installation and Migration)	COEXIST behavior is used even when YES is specified. That is, no automatic bind occurs in release coexistence or fallback scenarios when a	No

Subsystem parameter	Change introduced	Incompatible change?
	plan or package that was automatically bound in Db2 11 runs again in Db2 12. This change prevents repeating automatic remigration binds each time that a plan or package runs again on a different release during coexistence or fallback scenarios. For more information, see Automatic binds in coexistence (Db2 Installation and Migration).	
ALLOW_UPD_DEL_INS_WITH_U R in macro DSN6SPRM (Db2 Installation and Migration)	New subsystem parameter.	No
ALTERNATE COPY POOL field (ALTERNATE_CP subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
APPL COMPAT LEVEL field (APPLCOMPAT subsystem parameter) (Db2 Installation and Migration)	The default value for new installations is changed to 'V12R1M500'. Support is also added for values that correspond to Db2 12 function levels.	No
CDDS_MODE in macro DSN6LOGP (Db2 Installation and Migration)	New subsystem parameter.	No
CDDS_PREFIX in macro DSN6LOGP (Db2 Installation and Migration)	New subsystem parameter.	No
MINUTES/CHECKPOINT field (CHKFREQ and CHKMINS subsystem parameters) (Db2 Installation and Migration)	The default value is changed to 3 and the accepted range for the MINUTES/CHECKPOINT value is now 1 - 5 minutes. If the CHKTYPE subsystem parameter setting is SINGLE and the CHKFREQ subsystem parameter setting is 6 - 60, Db2 uses 5 for the CHKFREQ setting, issues a warning message, and sets RC=4 for subsystem parameter assembly. If the CHKTYPE subsystem parameter setting is BOTH and the CHKMINS subsystem parameter setting is 6 - 1439, Db2 uses 5 for the CHKMINS setting, issues a warning message, and sets RC=4 for subsystem parameter assembly.	No
COPY FAST REPLICATION field (COPY_FASTREPLICATION subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
DDF_COMPATIBILITY in macro DSN6FAC (Db2 Installation and Migration)	A new value is supported: RESET_IFCID402 specifies that Db2 resets IFCID 0402 counters when the associated profile threshold changes.	No
DEFAULT INSERT ALGORITHM field	• The default value is changed from 2 to 1.	No

Subsystem parameter	Change introduced	Incompatible change?
(DEFAULT_INSERT_ALGORITHM subsystem parameter) (Db2 Installation and Migration)	• A new value is added, 0, which means that the basic insert algorithm (insert algorithm 1) is used regardless of the specification at the object level.	
DISALLOW_SEL_INTO_UNION in macro DSN6SPRM (Db2 Installation and Migration)	The default is changed from NO to YES.	No
DISALLOW_SSARAUTH in macro DSN6SPRM (Db2 Installation and Migration)	New subsystem parameter.	No
EDM SKELETON POOL SIZE field (EDM_SKELETON_POOL subsystem parameter) (Db2 Installation and Migration)	The default value is changed to 51 200.	No
FLASHCOPY XRCP field (FLASHCOPY_XRCP subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
HSM MESSAGE DS HLQ field (UTILS_HSM_MSGDS_HLQ subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
INDEX MEMORY CONTROL field (INDEX_MEMORY_CONTROL subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter. (SELECTED,AUTO) and (SELECTED,n) settings are added to support enabling fast index traversal only for specific indexes. These settings are used when you specify YES in the FOR SELECTED INDEXES ONLY field on panel DSNTIP71.	No
INLISTP in macro DSN6SPRM (Db2 Installation and Migration)	The default value is changed to 1000.	No
LA_SINGLESEL_ISOCS_CDY subsystem parameter (Db2 Installation and Migration)	New subsystem parameter.	No
LOAD FORMAT DELIMITED IMPLICIT DECIMAL SCALE (LOAD_DEL_IMPLICIT_SCALE subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
LOAD RO OBJECTS field (LOAD_RO_OBJECTS subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
MFA AUTH UNUSED TIME field (MFA_AUTHCACHE_UNUSED_TIME subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No

Subsystem parameter	Change introduced	Incompatible change?
RID POOL SIZE field (MAXRBLK subsystem parameter) (Db2 Installation and Migration)	The default value is changed to 1000000 (1,000,000 kilobytes).	No
MAX UDFS field (MAX_UDF subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	Yes
NPGTHRSH in macro DSN6SPRM (Db2 Installation and Migration)	The default value is changed to 1, which means that Db2 favors matching index access, through the index with the most matching columns, for tables or partitions that have fewer than one data page. (For SAP installations, the default value remains 10.)	No
RO SWITCH TIME field (PCLOSET subsystem parameter) (Db2 Installation and Migration)	The default value is changed to 45.	No
PEER RECOVERY field (PEER_RECOVERY subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
PRIMARY QUANTITY field (PRIQTY subsystem parameter) (Db2 Installation and Migration)	The default value is changed to 125 (in cylinders —ACLUNIT is removed).	No
PROFILE AUTOSTART field (PROFILE_AUTOSTART subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
MANAGE REAL STORAGE field (REALSTORAGE_MANAGEMENT subsystem parameter) (Db2 Installation and Migration)	APAR PH47163 improves the behavior when the default setting AUTO is used, and introduces a new value AUTO1, which can be used to revert to the previous AUTO behavior if necessary. For more information, see APAR PH47163.	No
REORG TS NOPAD DEFAULT (REORG_TS_NOPAD_DEFAULT subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
REORG PART SORT NPSI field (REORG_PART_SORT_NPSI subsystem parameter) (Db2 Installation and Migration)	The behavior of the YES setting is changed so that Db2 always sorts the keys for non-partitioned secondary indexes (NPSIs). To continue with the previous behavior, use AUTO.	Yes
RLF SCOPE field (RLFENABLE subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
REMOTE STATIC SQL field (RLFERRDSTC subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No

Subsystem parameter	Change introduced	Incompatible change?
STATIC SQL field (RLFERRSTC subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
STATISTICS COLGROUP DATA SORT STG LIMIT field (STATCLGSRT subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
STATISTICS PROFILE FEEDBACK field (STATFDBK_PROFILE subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
DDF STATS TIME field (STATIME_DDF subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
MAIN STATS TIME field (STATIME_MAIN subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter. APAR PH56228 removes the control of the interval of location statistics trace records, IFCIDs 365, 411, and 412 from STATIME_MAIN.	No
PAGE-LEVEL SAMPLING field (STATPGSAMP subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
PREVENT_NEW_IXCTRL_PART in macro DSN6SPRM (Db2 Installation and Migration)	The default value is changed to YES.	No
SECONDARY QTY field (SECQTY subsystem parameter) (Db2 Installation and Migration)	The default value is changed to 15 (in cylinders—ACLUNIT is removed).	No
SMF ACCOUNTING field (SMFACCT subsystem parameter) (Db2 Installation and Migration)	The default value is changed to '1,2,3,7,8'.	No
DB BACKUP STG GROUP field (UTIL_DBBSG subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
LOG BACKUP STG GRP field (UTIL_LGBSG subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
UTILS USE ZSORT (UTILS_USE_ZSORT subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No

Subsystem parameter	Change introduced	Incompatible change?
DB2 zHyperLinks SCOPE field (ZHYPERLINK subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
ACCELERATION OPTIONS field (QUERY_ACCEL_OPTIONS subsystem parameter) (Db2 Installation and Migration)	New field on panel DSNTIP8A: 13 ENABLE ACCELERATOR SPECIFIC RESULTS	No

For more information about subsystem parameter changes in Db2 12, see "Subsystem parameter changes in Db2 12" on page 200.

Activation details for function level 100

After migration from Db2 11, Db2 subsystems and data sharing members are at function level 100. The Db2 catalog level is V12R1M500. Subsystems and data sharing members at function level 100 can coexist with Db2 11 new-function mode.

After the activation of function level 500 you can return a subsystem or group member to function level 100*. You can also activate function level 100* in new Db2 12 installations. However, Db2 subsystems and data sharing members at function level 100* cannot coexist with or fall back to Db2 11.

Catalog level for function level 100

Db2 12 function levels 100 and 500 require catalog level V12R1M500. Catalog level V12R1M500 is the result when you install a new Db2 12 subsystem, or when you tailor the catalog during migration to Db2 12, as described in Migration step 19: Tailor Db2 12 catalog: DSNTIJTC (Db2 Installation and Migration).

For a summary of changes in Db2 12, see "Catalog changes in Db2 12" on page 179.

Related tasks

Activating Db2 12 new function at migration (Db2 Installation and Migration)
Adopting new capabilities in Db2 12 continuous delivery
In Db2 12, 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)

Chapter 3. Changes to plan for in Db2 12

Use this information when you are planning migration to Db2 12 and for planning to adopt new capabilities that Db2 12 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 12 or the activation of Db2 12 function levels. It also contains summaries of the changes that Db2 12 introduces, including separate sections for changes in the Db2 12 initial release and changes in higher function levels. It also summarizes function the Db2 12 no longer supports, and deprecated function that remains supported, but might be removed eventually.

Tip: Changes from many Db2 11 APARs are built-in when you migrate to Db2 13:

- Enhancements from many new-function APARs in Db2 11 are built-in when you migrate to Db2 12. If you do not apply the PTFs for such APARs in Db2 11, plan for these changes to take effect when you migrate to Db2 12.
- Also check for other Db2 11 service-stream APARs for Db2 11 that were available before general availability of Db2 12. If you did not already complete any required actions that are described in ++HOLD data for these APARs, make plans to do so before migration to Db2 12.

Related reference
New, changed, and deleted codes (Db2 Codes)
New, changed, and deleted messages (Db2 Messages)

Incompatible changes in Db2 12

Before migrating your Db2 11 subsystem to Db2 12, applying maintenance in Db2 12, 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 12 migration process, or before applying maintenance.

Related concepts
Function that Db2 12 no longer supports
If you are migrating to Db2 12 from Db2 11, be aware of the function that Db2 12 no longer supports.

Changes to plan for in Db2 12
Use this information when you are planning migration to Db2 12 and for planning to adopt new capabilities that Db2 12 introduces.

Related reference
Deprecated function in Db2 12
Certain features and capabilities that Db2 12 for z/OS supports are *deprecated*, meaning that their use is discouraged. Although they remain supported except as noted below in Db2 12, support is likely to be removed eventually.

Incompatible changes summary for function levels 501 and higher

Some Db2 12 function levels introduce incompatible changes to Db2 12 that you must plan to resolve before you activate Db2 12 function levels. Application programming and SQL incompatibilities must be resolved for specific application programs before they run with the corresponding higher application compatibility level.

This section describes incompatible changes that Db2 12 introduces when you activate function levels 501 and higher in Db2 12.

Function level 510 incompatible changes

Function level 510 introduces no incompatible changes.

Function level 509 incompatible changes

Function level 509 introduces no incompatible changes.

Function level 508 incompatible changes

Function level 508 introduces no incompatible changes.

Function level 507 incompatible changes

Function level 507 introduces no incompatible changes.

Function level 506 incompatible changes

Function level 506 introduces the following incompatible changes:

Changes in explicitly created table space behavior

When the application compatibility level is set to V12R1M506 or higher, the following changes become effective:

- Dropping a table that resides in an explicitly created universal table space no longer returns an error. Instead, the table space is implicitly dropped.
- Dropping an auxiliary table that resides in an explicitly created LOB table space no longer leaves the LOB table space in the database. Instead, the table space is implicitly dropped.

Actions required:

- Any existing applications that use the DROP TABLESPACE statement to drop a universal table space with a table can now be changed to use the DROP TABLE statement instead.
- If an auxiliary table is dropped, any applications or tools that expect the LOB table space to remain for reuse must be modified accordingly.

Function level 505 incompatible changes

Function level 505 introduces the following incompatible changes:

RUNSTATS does not update real-time statistics by default

Starting in function level 505, the RUNSTATS utility no longer updates real-time statistics (RTS) by default.

Action required: If you want RUNSTATS to update RTS, take one of the following actions:

- In the RUNSTATS statement, specify TABLESAMPLE SYSTEM NONE and do not specify SAMPLE.
- Set the STATPGSAMP subsystem parameter to NO and do not specify SAMPLE in the RUNSTATS statement.

Newly supported user-defined functions

If your environment contains existing user-defined functions with the same names and signatures as the new built-in functions, applications with unqualified references to the existing user-defined functions might start invoking the built-in function instead of the user-defined function in certain situations.

Action required: For information about avoiding such situations, see Ensuring that Db2 executes the intended user-defined function (Db2 Application programming and SQL).

Function level 504 incompatible changes

The following changes might impact your Db2 environment and applications that run at application compatibility level V12R1M504 or higher, such as the following changes:

- If your Db2 applications have unqualified references to existing user-defined functions, they might start invoking new built-in functions instead if the names and signatures match, in certain situations. For information about avoiding such situations, see Ensuring that Db2 executes the intended user-defined function (Db2 Application programming and SQL).

- If your Db2 environment contains existing columns or variables with names such as CURRENT_TIMEZONE or CURRENT_SERVER, you must modify your applications to delimit these names. For example, you can issue the following queries to identify columns and variables with these names. For columns, issue:

```
SELECT * FROM SYSIBM.SYSCOLUMNS
   WHERE NAME IN('CURRENT_SERVER','CURRENT_TIMEZONE');
```

For variables, issue:

```
SELECT * FROM SYSIBM.SYSVARIABLES
WHERE NAME IN('CURRENT_SERVER','CURRENT_TIMEZONE');
```

Based on the result, you can then check for dependencies, for example by querying the SYSIBM.SYSPACKDEP catalog table, to identify any static applications that depend on the nondelimited forms of these names.

- CREATE TABLESPACE statements that previously created non-UTS segmented or partitioned table spaces, now always create partition-by-growth or partition-by-range table spaces.

  Tip: Although it is best to use the non-deprecated types for new objects, you can still use a lower application compatibility level to create the deprecated table spaces or higher if necessary, such as for recovery situations. For instructions, see Creating non-UTS table spaces (deprecated) (Db2 Administration Guide).

- Errors can be returned for SQL statements that previously succeeded, as side effects of the tables now being created in universal table spaces instead of non-UTS types. For example:

  - CREATE AUX TABLE statements for LOB columns can start failing with SQLCODE -769 because the PART clause is required when the base table resides in a PBG table space.

  - DROP TABLE statements can start failing with SQLCODE -669 because the DROP TABLE statement is not supported for tables in PBG table spaces. The DROP TABLESPACE statement must be used instead. (Function level 506 resolves this situation. See "Support for implicitly dropping explicitly created UTS and LOB table spaces" on page 82.)

  - CREATE TABLE statements can start failing with SQLCODE -646 if they attempt to create more than one table in a table space. Each universal table space contains only single table.

- The following SQL statements now return errors:

  - CREATE SYNONYM statements

  - CREATE TABLE statements that specify existing non-UTS table spaces

  - CREATE TABLE and ALTER TABLE statements that specify ORGANIZE BY HASH

  Action to take: Update your applications to use the non-deprecated object types.

Function level 503 incompatible changes

Function level 503 introduces the following incompatible changes.

Result change for system-period temporal tables defined with ON DELETE ADD EXTRA ROW

A query has incompatible behavior at application compatibility level V12R1M503 if all of the following conditions are met:

- It references a system-period temporal table (STT) with FOR SYSTEM_TIME FROM value1 TO value2 (or BETWEEN value1 AND value2) period specification. The STT can be directly referenced or indirectly referenced through views.
- The STT is defined with the ON DELETE ADD EXTRA ROW clause.
- The STT is defined with the DATA CHANGE OPERATION column.
- During migration to use the temporal auditing feature, the altered table contains existing rows.
- The DATA CHANGE OPERATION column is populated with the default NULL value for the existing rows. When the query is executed, the historical table contains records with a NULL value in DATA CHANGE OPERATION column.

Action required: Start a trace for IFCID 0376. In the trace output, function code 1215031 or 1202 identifies the affected queries.

The SET_MAINT_MODE_RECORD_NO_TEMPORALHISTORY stored procedure is not supported when APPLCOMPAT is V12R1M503 or higher

Users will not be able to invoke the SYSPROC.SET_MAINT_MODE_RECORD_NO_TEMPORALHISTORY stored procedure when running with APPLCOMPAT=V12R1M503 or higher. Existing applications that invoke the SYSPROC.SET_MAINT_MODE_RECORD_NO_TEMPORALHISTORY stored procedure will fail starting with APPLCOMPAT=V12R1M503 or higher.

Action required: Start a trace for IFCID 0376. In the trace output, function code 1215032 or 1203 identifies applications that use the SYSPROC.SET_MAINT_MODE_RECORD_NO_TEMPORALHISTORY stored procedure. Modify such applications to set value of the SYSIBMADM.REPLICATION_OVERRIDE built-in global variable to 'Y' instead. For more information, see SYSIBMADM.REPLICATION_OVERRIDE built-in global variable.

Function level 502 incompatible changes

If your environment contains existing user-defined functions named GRAPHIC or VARGRAPHIC, applications with unqualified references to the existing user-defined functions might start invoking the built-in function instead of the user-defined function in certain situations. For information about avoiding such situations, see Ensuring that Db2 executes the intended user-defined function (Db2 Application programming and SQL).

Function level 501 incompatible changes

Function level 501 introduces no incompatible changes.

Related concepts
Changes to plan for in Db2 12
Use this information when you are planning migration to Db2 12 and for planning to adopt new capabilities that Db2 12 introduces.

Related tasks
Adopting new capabilities in Db2 12 continuous delivery
In Db2 12, function levels and application compatibility levels control the adoption of most new capabilities by Db2 subsystems and Db2 applications.

Related reference
Db2 12 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.

Application and SQL release incompatibilities

When you migrate to or apply maintenance in Db2 12, be aware of and plan for application and SQL release incompatibilities that might affect your Db2 environment.

[GUPI]

The following incompatible changes apply at any Db2 12 function level, including when you first migrate to Db2 12. For incompatible changes that might impact your Db2 12 environment when you activate function levels 501 and higher, see "Incompatible changes summary for function levels 501 and higher" on page 119.

SQL capabilities

New SQL capabilities become available after the activation of the function level that introduces them or higher, for applications that run at the equivalent application compatibility level or higher. New SQL capabilities in the initial Db2 12 release become available in function level 500 for applications that run at the equivalent application compatibility level or higher. You can continue to run SQL statements compatibly with lower function levels, or previous Db2 releases, including Db2 11 and DB2 10. For details, see Application compatibility levels in Db2 12 (Db2 Application programming and SQL)

Release incompatibilities that were changed or added since the first edition of this Db2 12 publication are indicated by a vertical bar in the left margin. In other areas of this publication, a vertical bar in the margin indicates a change or addition that has occurred since the Db2 11 release of this publication.

SUBSTR built-in function always returns an error message for invalid input

Previously, during execution of the SUBSTR built-in function, Db2 sometimes incorrectly returned a result for invalid input instead of issuing an appropriate error message. After the PTF for APAR PH36071 is applied and Db2 12 function level 500 or higher is activated, the SUBSTR_COMPATIBILITY subsystem parameter is set to PREVIOUS by default and Db2 continues to behave as before the PTF was applied. If the SUBSTR_COMPATIBILITY subsystem parameter is set to CURRENT, Db2 always enforces the rules for the SUBSTR built-in function that are documented in the *SQL Reference* and returns an SQL error code if the rules are not met.

For example, if the SUBSTR_COMPATIBILITY subsystem parameter is set to CURRENT, the following query returns an SQL error code:

```
SELECT SUBSTR('ABCD', 2+1, 3) FROM SYSIBM.SYSDUMMY1;
```

Previously, this query incorrectly returned the result 'CD '.

Before you set the SUBSTR_COMPATIBILITY subsystem parameter to CURRENT, you might need to modify some of your applications to handle this change.

Actions to take

In Db2 12, before you set the SUBSTR_COMPATIBILITY subsystem parameter to CURRENT, identify applications that are incompatible with this change by starting a trace for IFCID 0376 and then running the applications. Review the trace output for incompatible changes with the identifier 14. Correct affected applications so that they will be compatible if the SUBSTR_COMPATIBILITY subsystem parameter is set to CURRENT in the future.

Related reference

SUBSTR COMPATIBILITY field (SUBSTR_COMPATIBILITY subsystem parameter) (Db2 Installation and Migration)

CREATE TABLESPACE and CREATE INDEX statements with no space-level USING clause fail if the storage group specified when the containing database was created does not exist

Starting in Db2 12 at function level 500, Db2 records the default storage group for a table space or index in the Db2 catalog. However, Db2 also does not validate the existence of a storage group specified in

the STOGROUP clause of a CREATE DATABASE statement. As a result, a CREATE TABLESPACE or CREATE INDEX statement that omits the USING clause at the table space or index level now fails with SQLCODE -204, if the storage group specified when the containing database was created does not exist.

Actions to take

If the storage group specified in a CREATE DATABASE statement does not exist, take one of the following actions:

- Specify a USING clause at the table space or index level in any CREATE TABLESPACE or CREATE INDEX statement that creates a table space or index in that database.
- Issue an ALTER DATABASE statement and specify STOGROUP clause that identifies a storage group that exists.

Related reference
CREATE TABLESPACE (Db2 SQL)
CREATE INDEX (Db2 SQL)
CREATE DATABASE (Db2 SQL)
ALTER DATABASE (Db2 SQL)

New maximum number of parameter markers or host variables in a single SQL statement

Db2 12 enforces the maximum number of parameter markers or host variables in a single SQL statement. Starting in function level 100, Db2 12 issues SQLCODE -101 for any SQL statement that contains more than 16,000 parameter markers or host variables.

Actions to take

Identify and modify any existing SQL statement that contains more than 16,000 parameter markers or host variables.

Related reference
Limits in Db2 for z/OS (Db2 SQL)

Result change for SQL statement EXPLAIN PACKAGE

When Db2 processes the SQL statement EXPLAIN PACKAGE, the HINT_USED column in the PLAN_TABLE is populated with EXPLAIN PACKAGE: *copy*. The *copy* field in the HINT_USED column will be one of the following values:

- "CURRENT" - the current copy
- "PREVIOUS" - the previous copy
- "ORIGINAL" - the original copy

This change supports the new rebind phase-in capability that is introduced by function level 505. However, the change takes effect immediately when you migrate to Db2 12.

Actions to take

Change the expected output for queries that reference this column.

Result change for SQL statement EXPLAIN STABILIZED DYNAMIC QUERY

When Db2 processes the SQL statement EXPLAIN STABILIZED DYNAMIC QUERY, the HINT_USED column in the PLAN_TABLE is populated with EXPLAIN PACKAGE: *copy*. The *copy* field in the HINT_USED column will be one of the following values:

- "CURRENT" - the current copy
- "INVALID" - the invalid copy

This change supports the new rebind phase-in capability that is introduced by function level 505. However, the change takes effect immediately when you migrate to Db2 12.

Actions to take

Change the expected output for queries that reference this column.

SYSCOPY catalog table DSVOLSER column changes

Db2 12 introduces a new capability to delete only FlashCopy image copies if equivalent sequential image copies exist, for an efficient backup procedure that uses minimal disk space. In support of this capability, the possible values for the DSVOLSER column in the SYSIBM.SYSCOPY catalog table have changed. Previously, the DSVOLSER column value was an empty string for cataloged, sequential, full image copies. Some applications might assume that if the length attribute of the DSVOLSER value is zero, the image copy is cataloged. In Db2 12, that assumption is no longer correct. For cataloged, sequential, full image copies that are created from a FlashCopy image copy with consistency, and also had uncommitted units of work backed out, the DSVOLSER column now contains Db2 checkpoint information.

For more information about this new capability, see Ability to delete only FlashCopy image copies.

Actions to take

Modify any applications that use the DSVOLSER column in the SYSCOPY catalog table to tolerate the checkpoint information for cataloged, sequential, full image copies. For details, see the description of DSVOLSER in SYSCOPY catalog table.

Application compatibility levels apply to data definition and data control statements

After the activation of function level 500 or higher in Db2 12, application compatibility levels also control syntax, semantics, default values, and option validation for most data definition statements and data control statements. Data definition statements (sometimes abbreviated as DDL) include various CREATE and ALTER statements. Data control statements (sometimes abbreviated as DCL) include various GRANT and REVOKE statements. Only application compatibility levels V12R1M509 and higher control the behavior of any data definition or data control statements.

The APPLCOMPAT bind option for a package applies to most static SQL data definition and data control statements. The CURRENT APPLICATION COMPATIBILITY special register applies to most dynamic SQL data definition and data control statements.

For implicit regeneration of an object, the application compatibility level that was in effect for the previous CREATE or ALTER statement for that object is used.

For materialization of pending data definition changes, the application compatibility level of the pending ALTER statement is used.

You can specify the USING APPLICATION COMPATIBILITY clause of certain ALTER statements to regenerate an object with a specific application compatibility level.

Related concepts
Function levels and related levels in Db2 12
Enhancements to Db2 12 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 12 (Db2 Application programming and SQL)
Related reference
APPLCOMPAT bind option (Db2 Commands)
CURRENT APPLICATION COMPATIBILITY special register (Db2 SQL)

Automatic rebind of plans and packages created before DB2 10

Migration-related *automatic binds* (also called "autobinds") occur in Db2 12 because it cannot use runtime structures from a plan or package that was last bound in a release earlier than DB2 10. Plans and packages that were bound in Db2 11 can run in Db2 12, without the risk of migration-related autobinds. However, plans and packages that are bound in Db2 12 cannot run on Db2 11 members without an autobind in Db2 11.

If you specify YES or COEXIST for the ABIND subsystem parameter, Db2 12 automatically rebinds plans and packages that were bound before DB2 10 when Db2 executes the packages. The result of the automatic bind creates a new package and discards the current copy. Db2 does not move the current copy to the previous or original copy because Db2 12 cannot use it. If a regression occurs, REBIND SWITCH PREVIOUS and REBIND SWITCH ORIGINAL are not available.

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 10. SQLCODE -908, SQLSTATE 23510 is returned for packages, and SQLCODE -923, SQLSTATE 57015 is returned for plans until they are rebound in Db2 12.

Actions to take

By preparing for migration to Db2 12 in Db2 11, you can reduce the change and risk for packages that are subject to automatic binds in Db2 12. To do that, you rebind all packages that were last bound before DB2 10 in Db2 11, before you migrate to Db2 12. 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 11 to avoid disruptive autobinds in Db2 12 (Db2 Installation and Migration).

KEEPDYNAMIC(YES) bind option support for ROLLBACK

In Db2 12, when the APPLCOMPAT value is V12R1M500, the KEEPDYNAMIC(YES) bind option affects both COMMIT and ROLLBACK statements. With KEEPDYNAMIC(YES), the dynamic SQL statements in the package are retained after COMMIT or ROLLBACK, and those statements can run again without another PREPARE.

Prior to Db2 12, the KEEPDYNAMIC(YES) bind option applied only to COMMIT statements. After a ROLLBACK statement, another PREPARE was required so that the dynamic SQL statements could run. This situation is also true in Db2 12 if application compatibility is set to V11R1 or earlier.

In Db2 12, when the APPLCOMPAT value is V12R1M500 or higher, after a ROLLBACK statement is issued, the behavior is different than in prior versions:

- An OPEN statement without a preceding PREPARE statement does not receive an SQLCODE -514.
- An EXECUTE statement without a preceding PREPARE statement does not receive an SQLCODE -518.

An application that was written in Db2 11 and that was bound with KEEPDYNAMIC(YES) was required to prepare dynamic SQL statement again after a ROLLBACK was issued. In Db2 12 when application compatibility is set to V12R1M500 or higher, those extra PREPARE statements are unnecessary.

Actions to take

As you migrate to Db2 12, review packages that use the KEEPDYNAMIC(YES) bind option. You can make dynamic SQL programs that are bound with KEEPDYNAMIC(YES) run more efficiently by removing PREPARE statements that prepare SQL statements again, following execution of ROLLBACK statements. Do not take this action until you are sure that you no longer need to run the programs in Db2 11 or earlier. After migrating to Db2 12, if you take this action (to remove PREPARE statements after ROLLBACK), programs will not work properly if you subsequently set application compatibility to V11R1 or earlier.

Alterations to index compression are pending changes for universal table spaces

When the application compatibility level is V12R1M500 or higher, altering to use index compression for indexes in universal table spaces is a pending change that places the index in advisory REORG-pending (AREOR) status. The LOAD REPLACE and REBUILD INDEX utilities no longer materialize the change. You

must use an online REORG to materialize the new value for the COMPRESS attribute in the ALTER INDEX statement.

In releases before Db2 12, any alteration to use index compression placed the index in REBUILD-pending (RBDP) status. You needed to use the REBUILD INDEX utility to rebuild the index, or use the REORG utility to reorganize the table space that corresponds to the index.

Actions to take

For indexes in universal table spaces, use an online REORG to materialize the new value for the COMPRESS attribute in the ALTER INDEX statement.

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

In function level 500 or higher with application compatibility set to V11R1, when a Java™ application that uses the IBM Data Server Driver for JDBC and SQLJ calls a stored procedure, the data types of stored procedure output arguments match the data types of the parameters in the stored procedure definition.

Explanation

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

In Db2 12, when application compatibility is set to V10R1, you can set the DDF_COMPATIBILITY subsystem parameter to SP_PARMS_JV to keep the behavior that existed before DB2 10. However, when application compatibility is set to V11R1 or V12R1M100, or to V12R1M500 or higher, SP_PARMS_JV is no longer supported.

In Db2 12 with application compatibility set to V11R1 or V12R1M100, or to V12R1M500 or higher, if the version of the IBM Data Server Driver for JDBC and SQLJ is lower than 3.63 or 4.13, a `java.lang.ClassCastException` might be thrown when an output argument value is retrieved.

Actions to take

Take one of the following actions:

- Upgrade the IBM Data Server Driver for JDBC and SQLJ to version 3.63 or 4.13, or later.
- Modify the data types in `CallableStatement.registerOutParameter` method calls to match the parameter data types in the stored procedure definitions. You can set application compatibility to V10R1 and run a trace for IFCID 0376 to identify affected applications. Trace records for those applications have a QW0376FN field value of 8.

Related concepts

Application compatibility levels in Db2 12 (Db2 Application programming and SQL)

SELECT INTO statements with UNION or UNION ALL

A UNION or UNION ALL is not allowed in the outermost from-clause of a SELECT INTO statement. However, releases before Db2 12 inadvertently tolerate SQL statements that contain this invalid syntax.

The behavior is controlled by the DISALLOW_SEL_INTO_UNION subsystem parameter. In Db2 11, the default setting tolerates the invalid syntax. In Db2 12 the default setting disallows the invalid syntax.

An application that uses the invalid SQL syntax fails at BIND or REBIND with SQLCODE -109.

Actions to take

Identify any packages that use UNION or UNION ALL in the from-clause of a SELECT INTO statement and correct them as necessary. You can temporarily specify that Db2 continues to tolerate the invalid syntax

NO for the DISALLOW_SEL_INTO_UNION subsystem parameter. However, this subsystem parameter is deprecated and expected to be removed in the future.

You can identify affected packages while DISALLOW_SEL_INTO_UNION is set to NO by binding suspected packages into a dummy collection ID with EXPLAIN(ONLY) and monitoring IFCID 0376 records. Trace records for the affected applications have a QW0376FN field value of 11.

Use the following procedure:

1. Issue the following SQL statement to generate a list of BIND commands.

```
SELECT 'BIND PACKAGE(DUMMYCOL) COPY(' ||
       COLLID || '.' || NAME || ') ' ||
       CASE WHEN(VERSION <> '')
            THEN 'COPYVER(' || VERSION || ') '
            ELSE '' END ||
       'EXPLAIN(ONLY)'
FROM SYSIBM.SYSPACKSTMT
 WHERE STATEMENT LIKE '%SELECT%INTO%UNION%'
 OR STATEMENT LIKE '%SELECT%UNION%INTO%';
```

The statement generates output similar to the following BIND subcommand:

```
BIND PACKAGE(DUMMYCOL) COPY(DSN_DEFAULT_COLLID_PLAY01.PLAY01) EXPLAIN(ONLY)
```

2. Copy the results of the SELECT statement into a bind job. If any BIND subcommands are longer than 72 bytes, formatting is required.

3. Start and collect a trace for IFCID 0376.

4. Run the bind job that you created.

5. Stop the IFCID 0376 trace and analyze the output.

Related reference
DISALLOW_SEL_INTO_UNION in macro DSN6SPRM (Db2 Installation and Migration)
Related information
-109 (Db2 Codes)

Change in SQLCODE when the POWER built-in function result is out of range

After the activation of function level 500 or higher in Db2 12, the SQLCODE that is returned when the result of the POWER® built-in function is out of range is changed in some cases.

Previously, when Db2 executed the POWER built-in function, and the result was a DOUBLE data type that was out of range, Db2 returned SQLCODE -802. In Db2 12 with function level 500 or higher activated, SQLCODE +802 is returned.

For example, the following query returns SQLCODE +802:

```
SELECT POWER(DOUBLE(2.0E38), DOUBLE(2.0))
 FROM SYSIBM.SYSDUMMY1;
```

Invocations of the POWER function that have DOUBLE arguments and return out-of-range results return SQLCODE +802 instead of SQLCODE -802.

Actions to take

In Db2 12, before function level 500 or higher is activated, identify applications with this incompatibility by starting a trace for IFCID 0376, and then running the applications. Review the trace output for incompatible changes with the identifier 1201. Adjust error processing to account for the change in the returned SQLCODE from an error to a warning.

Related tasks
Managing application incompatibilities (Db2 Application programming and SQL)

Related reference

POWER or POW (Db2 SQL)

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

DB2 10 changed the formatting of decimal data by the CHAR and VARCHAR built-in functions and CAST specifications with a CHAR or VARCHAR result type. In Db2 12 you can use alternative functions for compatibility with applications that require decimal to string output in the pre-DB2 10 formats:

- CHAR9 (Db2 SQL)

- VARCHAR9 (Db2 SQL)

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

Actions to take

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

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

1. Use an IFCID 0376 trace to identify applications that depend on the pre-DB2 10 formats.
2. Rewrite the SQL statements in the identified applications to use the CHAR9 and VARCHAR9 functions instead of the CHAR and VARCHAR functions.
3. Set the BIF_COMPATIBILITY value to CURRENT.

Related reference

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

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

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

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

Actions to take

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

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

   To handle the change for the CHAR function only, you can set BIF_COMPATIBILITY to V9, and complete the following steps for the CHAR function.
3. Change any affected applications to handle the DB2 10 and later CHAR and VARCHAR behavior, including stored procedures, non-inline user-defined functions, and trigger packages. Rewrite affected CAST specifications with the appropriate CHAR or VARCHAR function and a CAST to the correct length if needed.

4. Rebind and prepare packages with the PATH(SYSCURRENT,SYSIBM) rebind option. Putting the SYSCURRENT schema at the beginning of the SQL path causes the DB2 10 and later behavior to be used for the CHAR and VARCHAR built-in functions.

   Repeat this step for native stored procedures (SQLPL) and non-inline SQL scalar functions.

5. For views that reference these casts or built-in functions, determine whether the view needs to be changed to have the expected output. Drop and re-create the views with the PATH(SYSCURRENT,SYSIBM) rebind option only if necessary. Rebind any applications that reference the views with the PATH(SYSCURRENT,SYSIBM) option to use the DB2 10 and later CHAR and VARCHAR built-in functions. Repeat this step for inline SQL scalar functions.

6. For materialized query tables or indexes on expressions that reference these casts or built-in functions, drop and re-create the materialized query tables or indexes on expressions with the PATH(SYSCURRENT,SYSIBM) rebind option. Issue the REFRESH TABLE statement for materialized query tables. Rebind any applications that reference the materialized query tables or indexes on expressions with the PATH(SYSCURRENT,SYSIBM) option to use the DB2 10 and later CHAR and VARCHAR built-in functions.

7. Change the value of the BIF_COMPATIBILITY subsystem parameter to CURRENT. When the subsystem parameter value is CURRENT, new applications, rebinds, and CREATE statements use the DB2 10 and later CHAR, VARCHAR, and CAST behavior.

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

Related reference

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

EBCDIC mixed string input to the RTRIM, TRIM, LTRIM, and STRIP built-in functions must be valid

Starting at application compatibility level V12R1M500 or higher, Db2 12 applies more validation checking for EBCDIC mixed-string input to the RTRIM, TRIM, LTRIM, and STRIP built-in functions.

Generally, Db2 has required valid EBCDIC mixed-string data for input to these functions since version 10, but Db2 12 now detects more cases than earlier releases.

With the new validation checking, when Db2 12 performs a trim operation, and the *string-expression* argument of the RTRIM, TRIM, LTRIM, or STRIP built-in function contains invalid EBCDIC mixed data, Db2 issues SQLCODE -171.

Actions to take

Check whether EBCDIC mixed data that is specified for the *string-expression* argument of an RTRIM, TRIM, LTRIM, or STRIP built-in function is valid, and resolve the invalid data.

In valid mixed data, the double-byte portions of the input strings begin with X'0E' (shift-out character), end with X'0F' (shift-in character), and have an even number of bytes between the X'0E' and X'0F' characters. Data that does not meet these criteria is invalid mixed data. When invalid mixed data is specified for the *string-expression* argument of an RTRIM, TRIM, LTRIM, or STRIP built-in function, SQLCODE -171 is returned in some cases. If Db2 trims the specified characters before it reaches an invalid portion of a mixed string, the trim operation is successful.

Setting an application compatibility of V12R1M100 or lower (if PTF for APAR PH25783 is applied) might avoid the error while you resolve the invalid data. At the lower application compatibility levels, Db2 12 attempts to tolerate the invalid mixed data identified by the new validation checking, and it allows the trim operation to complete if possible. However, if Db2 is still unable to perform the trim operation, SQLCODE -171 is issued.

Maximum number of user-defined external scalar functions running in a Db2 thread is no longer unlimited (APAR PH44833)

Starting at application compatibility level V12R1M100 (if the PTF for APAR PH44833 is applied), Db2 12 introduces the MAX_UDF subsystem parameter. MAX_UDF 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.

Actions to take

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.

Related reference
MAX UDFS field (MAX_UDF subsystem parameter) (Db2 Installation and Migration)
Related information
00E70082 (Db2 Codes)

SQL reserved words

`PSPI`

Db2 12 introduces several new SQL reserved words, which are listed in Reserved words in Db2 for z/OS (Db2 SQL).

In some cases, the use of these reserved words might cause an incompatibility before new function is activated in Db2 12 , regardless of the setting of the APPLCOMPAT flag.

Actions to take
Collect IFCID 0376 trace records in Db2 11. Values 4, 5, and 6 for the QW0376FN field indicate instances of reserved words in applications that will cause an incompatibility in Db2 12. Adjust these applications by changing the reserved word to a delimited identifier or by using a word that is not reserved in Db2 12. `PSPI`

Built-in function and existing user-defined functions

For built-in and user-defined functions the combination of the function name and the parameter list form the *signature* that Db2 uses to identify the function. If the signatures of new or changed built-in functions in Db2 12 match the signatures existing user-defined functions, applications with unqualified references to the existing user-defined functions might start invoking the new or changed built-in functions instead of the user-defined functions. Db2 12 introduces the following built-in function changes:

Db2 12 introduces or changes the following built-in functions.

Important information about existing user-defined functions: When a new application compatibility level introduces a new or changed built-in function that has the same name and signature as an existing user-defined function, unqualified references to the user-defined function might resolve incorrectly. Applications that have unqualified references to the user-defined function might fail. To avoid this situation, modify applications to explicitly qualify references to user-defined functions with the same name and signature as the new or changed built-in functions.

`GUPI`

APPLCOMPAT level	Function name	Change introduced	Incompatible change?
V12R1M507	Various	The following functions are newly supported in Db2 for z/OS as passthrough-only expressions, which are passed through to IBM Db2 Analytics Accelerator for z/OS. • ADD_DAYS (Db2 SQL) • BTRIM (Db2 SQL) • DAYS_BETWEEN (Db2 SQL) • NEXT_MONTH (Db2 SQL) • REGR_AVGX • REGR_AVGY • REGR_COUNT • REGR_INTERCEPT or REGR_ICPT • REGR_R2 • REGR_SLOPE • REGR_SXX • REGR_SXY • REGR_SYY • ROUND_TIMESTAMP (Db2 SQL) if invoked with a date expression	No
V12R1M506	HASH (Db2 SQL)	New built-in function.	No
V12R1M506	CHARACTER_LENGTH or CHAR_LENGTH (Db2 SQL)	CHAR_LENGTH is now supported as an alternative function name.	No
V12R1M506	CLOB (Db2 SQL)	TO_CLOB is now supported as an alternative function name.	No
V12R1M506	COVAR_POP or COVARIANCE or COVAR (Db2 SQL)	COVAR_POP is now supported as an alternative function name.	No
V12R1M506	LEFT (Db2 SQL)	STRLEFT is now supported as an alternative function name.	No
V12R1M506	POWER or POW (Db2 SQL)	POW is now supported as an alternative function name.	No
V12R1M506	POSSTR (Db2 SQL)	STRPOS is now supported as an alternative function name.	No
V12R1M506	RANDOM or RAND (Db2 SQL)	RANDOM is now supported as an alternative function name.	No
V12R1M506	RIGHT (Db2 SQL)	STRRIGHT is now supported as an alternative function name.	No
V12R1M506	TIMESTAMP_FORMAT (Db2 SQL)	TO_TIMESTAMP is now supported as an alternative function name.	No
V12R1M505	DECRYPT_DATAKEY_INTEGER, DECRYPT_DATAKEY_BIGINT,	New built-in functions.	No

APPLCOMPAT level	Function name	Change introduced	Incompatible change?
	DECRYPT_DATAKEY_DECIMAL, DECRYPT_DATAKEY_VARCHAR, DECRYPT_DATAKEY_CLOB, DECRYPT_DATAKEY_VARGRAPHIC, DECRYPT_DATAKEY_DBCLOB, and DECRYPT_DATAKEY_BIT		
V12R1M505	ENCRYPT_DATAKEY	New built-in function.	No
V12R1M504	Various	The following functions are newly supported in Db2 for z/OS as passthrough-only expressions, which are passed through to IBM Db2 Analytics Accelerator for z/OS. • CUME_DIST • CUME_DIST (aggregate) (Db2 SQL)FIRST_VALUE • LAG • LAST_VALUE • LEAD • NTH_VALUE • NTILE • PERCENT_RANK • PERCENT_RANK (aggregate) (Db2 SQL) • RATIO_TO_REPORT • REGEXP_COUNT (Db2 SQL) • REGEXP_INSTR (Db2 SQL) • REGEXP_LIKE (Db2 SQL) • REGEXP_REPLACE (Db2 SQL) • REGEXP_SUBSTR (Db2 SQL)	No
V12R1M502	GRAPHIC (Db2 SQL)	The first argument now accepts numeric data types, including SMALLINT, INTEGER, BIGINT, DECIMAL, REAL, DOUBLE, FLOAT, and DECFLOAT.	No
V12R1M502	VARGRAPHIC (Db2 SQL)	The first argument accepts numeric data types, including SMALLINT, INTEGER, BIGINT, DECIMAL, REAL, DOUBLE, FLOAT, and DECFLOAT.	No
V12R1M501	LISTAGG (Db2 SQL)	New built-in function.	No
V12R1M500	ARRAY_AGG (Db2 SQL)	Newly supported built-in function when used for associative array aggregation.	No
V12R1M500	GENERATE_UNIQUE_BINARY	New built-in function.	No

APPLCOMPAT level	Function name	Change introduced	Incompatible change?
V12R1M500	HASH_CRC32, HASH_MD5, HASH_SHA1, and HASH_SHA256	New built-in functions.	No
V12R1M500	LOWER (Db2 SQL)	The following locales can now be specified: • UNI_60 • UNI_90	
V12R1M500	PERCENTILE_CONT	New built-in function.	No
V12R1M500	PERCENTILE_DISC	New built-in function.	No
V12R1M500	TRANSLATE (Db2 SQL)	The following locales can now be specified: • UNI_60 • UNI_90	No
V12R1M500	UPPER (Db2 SQL)	The following locales can now be specified: • UNI_60 • UNI_90	No
V12R1M500	WRAP (Db2 SQL)	New built-in function.	No
V12R1M100	BLOCKING_THREADS (Db2 SQL)	New built-in function.	

◁ GUPI

Actions to take

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

SQLCODE changes

Some SQLCODE numbers and message text might have changed in Db2 12. Also, the conditions under which some SQLCODEs are issued might have changed. For more information, see New, changed, and deleted codes (Db2 Codes).

◁ GUPI

Db2 utility release incompatibilities

When you migrate to or apply maintenance in Db2 12, be aware of the Db2 utility release incompatibilities that might affect your Db2 environment.

The following incompatible changes apply at any Db2 12 function level, including when you first migrate to Db2 12. For incompatible changes that might impact your Db2 12 environment when you activate function levels 501 and higher, see "Incompatible changes summary for function levels 501 and higher" on page 119.

Release incompatibilities that were changed or added since the first edition of this Db2 12 publication are indicated by a vertical bar in the left margin. In other areas of this publication, a vertical bar in the margin indicates a change or addition that has occurred since the Db2 11 release of this publication.

RECOVER utility changed to issue new messages for errors during index log processing

Starting in Db2 12 at function level 100, the RECOVER utility will issue messages DSNU1577I and DSNU1578I for errors encountered during index log processing. DSNU599I will no longer be issued in these cases.

Actions to take

If you have a process that scans for error messages in RECOVER utility output, you might need to change the process to scan for the message DSNU1578I, which is now issued to report log processing errors on indexes. You might also need to change the process to scan for the message DSNU1577I which is now issued by utilities to report indexes in page set REBUILD-pending status when a utility finishes.

Related reference
RECOVER (Db2 Utilities)
Related information
PH41586

New LOAD utility behavior change for identity columns

Starting in Db2 12 at function level 100, the LOAD utility always maintains the value of MAXASSIGNEDVAL in SYSIBM.SYSEQUENCES for any loaded identity columns.

Prior to Db2 12, the LOAD utility did not maintain the value of MAXASSIGNEDVAL when the identity column value was provided by the user. The identity column value can be provided by the user if the column is defined as GENERATED BY DEFAULT or if the column is defined as GENERATED ALWAYS and the OVERRIDE(IDENTITY) option is specified in the LOAD statement.

Actions to take

If your existing applications have logic that relies on the behavior prior to Db2 12, you can disable this new behavior by specifying UPDMAXASSIGNEDVAL NO in your LOAD statements. To aid your migration to Db2 12, you can begin adding this new option to your LOAD jobs in Db2 11. Although UPDMAXASSIGNEDVAL NO is accepted in Db2 11, the option has no effect until Db2 12.

Related concepts
Identity columns (Db2 Application programming and SQL)
Related reference
Syntax and options of the LOAD control statement (Db2 Utilities)

Changes to REORG TABLESPACE PART SORTNPSI YES behavior

Starting in Db2 12 at function level 100, when you specify the REORG TABLESPACE utility option SORTNPSI YES or set the REORG_PART_SORT_NPSI subsystem parameter to YES, REORG sorts the keys for all non-partitioned secondary indexes (NPSIs). Previously, when you specified these options, REORG sorted NPSIs only when certain internal thresholds were met. Therefore, some REORG SORTNPSI YES jobs that previously did not qualify for a sort operation will include a sort operation in Db2 12. Sorts can impact performance and might increase the elapsed time of those utility jobs.

Actions to take

If the extra sort operations cause a performance impact, set the subsystem parameter REORG_PART_SORT_NPSI to AUTO, or specify SORTNPSI AUTO in the REORG TABLESPACE utility statement.

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

Changes to the default behavior of REORG TABLESPACE and LOAD REPLACE

In Db2 12, the RRF subsystem parameter is removed. The default behavior is RRF=ENABLE, which means that:

- Newly created table spaces or newly added partitions on partition-by-growth table spaces are created in the reordered row format.
- If LOAD REPLACE or REORG TABLESPACE do not specify the ROWFORMAT BRF option, those utilities convert existing basic row format table spaces to reordered row format.

If the HONOR_KEEPDICTIONARY subsystem parameter is set to or defaults to NO, and LOAD REPLACE or REORG TABLESPACE converts an existing basic row format table space to reordered row format, a very large amount of sort work space is needed for decompression of the rows. The utilities fail if there is not enough sort work space available.

Actions to take

Set subsystem parameter HONOR_KEEPDICTIONARY to YES, and run LOAD REPLACE or REORG TABLESPACE with the KEEPDICTIONARY option. When you take these actions, Db2 does not decompress the data when the utilities run.

Related concepts
Row format conversion for table spaces (Db2 Administration Guide)
Related reference
Subsystem parameter changes in Db2 12
You can use this information to plan for subsystem parameter changes in Db2 12.

Changes to the mapping table for REORG

In Db2 12, the format of mapping tables has changed to accommodate the 7-byte RIDS that are needed for partition-by-range table spaces with relative page numbering. (Mapping tables are used by the REORG TABLESPACE utility when SHRLEVEL CHANGE is specified.) In the mapping table, the ORID column has changed to CHAR(7) instead of CHAR(5), and the NRID column has changed to CHAR(11) instead of CHAR(9).

After Db2 12 function level 500 or higher is activated, if you use the Db2 11 format for a mapping table, REORG TABLESPACE does not use it. Instead, the utility implicitly creates a mapping table in the Db2 12 format in the same database as the mapping table that you specified.

Actions to take

Either create your mapping tables according to the Db2 12 format or allow REORG to implicitly create the mapping tables for you.

Related concepts
Before running REORG TABLESPACE (Db2 Utilities)

Changes to the DSNACCOX stored procedure result set

In Db2 12, the DSNACCOX result set includes one or more new columns. This expanded result set is returned when the function level is function level 500 or higher. The result set includes the new column RRTPBGSPACEPCT. This column is reserved for future use.

Additionally, the result set always includes two columns that were optional in previous releases. In Db2 12, columns RRIEMPTYLIMIT and RRTHASHOVRFLWRAT are always returned when the function level is function level 500 or higher.

Applications that issue SELECT(*) against the DSNACCOX result set might fail or return unpredictable results.

Actions to take

Review any applications that process the results of the DSNACCOX stored procedure. If necessary, update them to handle the new result set.

Changes to allocation of image copy data sets on tape for RECOVER

In Db2 12, in RECOVER jobs, you can no longer include DD statements in the JCL to allocate image copy data sets that are on tape. In Db2 12, you must let RECOVER dynamically allocate image copy data sets that are on tape.

RECOVER jobs that contain DD statements for image copy data sets that are on tape fail with error DSNU030I.

Actions to take

If a RECOVER job fails with error DSNU030I, find DD statements for image copy data sets that are on tape, and remove the DD statements.

Changes to the behavior of the RESTORE SYSTEM utility

In Db2 12, RESTORE SYSTEM fails if either of the following situations occur:

- The most recent system-level backup before the recovery point is not registered in the DFSMShsm repository.
- An error is returned by DFSMShsm during the restore of the production volumes.

In previous releases, if RESTORE SYSTEM encountered a problem with the system-level backup, the utility automatically used the previous system-level backup. The new behavior in Db2 12 gives you control over the situation and allows you to determine the best course of action.

RESTORE SYSTEM jobs might fail.

Actions to take

Determine how you want to handle these failures. If you want to use an older system-level backup, use the new RESTOREBEFORE option in the RESTORE SYSTEM utility statement.

Changes to the invalidation of cached dynamic SQL statements by utilities that collect statistics

In Db2 12, the RUNSTATS utility and other utilities that collect inline statistics do not invalidate dynamic statements in the dynamic statement cache by default. In previous releases, the collection of statistics for an object always invalidated cached dynamic statements that referred to the objects. In Db2 12, the INVALIDATECACHE option is added to the RUNSTATS utility and to the following utilities that collect inline statistics:

- LOAD
- REBUILD INDEX
- REORG INDEX
- REORG TABLESPACE

The default option INVALIDATECACHE NO means that the utility does not invalidate statements in the dynamic statement cache because statistics are collected. Db2 might continue to use access paths that are based on stale statistics for dynamic SQL statements that remain in the dynamic statement cache.

Actions to take

Add the INVALIDATECACHE YES option to any existing utility jobs that collect statistics and must invalidate SQL statements in the dynamic statement cache.

Related tasks
Invalidating statements in the dynamic statement cache (Db2 Performance)
Related reference
Utility changes in Db2 12
You can use this information to plan for utility changes in Db2 12.

Changes to the behavior of the UNLOAD utility

In Db2 12, the UNLOAD utility includes the REGISTER option, which specifies whether, in a data sharing environment, pages that are read by the UNLOAD utility are registered with the coupling facility. REGISTER NO reduces data sharing overhead. In previous releases, the UNLOAD utility had only REGISTER YES behavior. In Db2 12, the original default for UNLOAD was REGISTER NO, but APAR PI99075 changes the default to REGISTER YES. REGISTER NO causes pages to be read directly from disk, and not from local or group buffer pools.

Actions to take

Apply APAR PI99075 to restore Db2 11 default behavior. Then update UNLOAD utility jobs with REGISTER NO, when appropriate to reduce data sharing overhead.

Changes to the authorization check of the UNLOAD utility

After the activation of function level 500 or higher, Db2 checks for the new UNLOAD privilege, instead of the SELECT privilege, for executing the UNLOAD utility and UNLOAD jobs might fail due to authorization errors.

Db2 checks the new UNLOAD privilege for authorization of the UNLOAD utility. You must use a privilege set that includes one of the following to execute the UNLOAD utility:

- Ownership of the tables
- UNLOAD privilege on the tables
- SELECT privilege on the tables (if the AUTH_COMPTIBILITY subsystem parameter is set to SELECT_FOR_UNLOAD)
- DBADM authority for the database. If the object on which the utility operates is in an implicitly created database, DBADM authority on DSNDB04 or the implicitly created database is sufficient.
- DATAACCESS authority
- SYSADM authority
- SYSCTRL authority (catalog tables only)
- SQLADM authority (catalog tables only)
- System DBADM authority (catalog tables only)
- ACCESSCTRL authority (catalog tables only)
- SECADM authority (catalog tables only)

When an UNLOAD job is run by a user with only the SELECT privilege on the target table, the UNLOAD job fails with message DSNU1253I.

Actions to take

Take any of the following actions:

- Before migration to Db2 12, run job DSNTIJPM, and follow the steps in premigration report 19 to determine which authorization IDs might need the UNLOAD privilege.
- In Db2 11 or Db2 12, start a trace for IFCID 404 to identify the users who need the UNLOAD privilege. Then, in Db2 12 at function level 100, grant the UNLOAD privilege to those users.

- As a temporary solution, set the AUTH_COMPATIBILITY subsystem parameter to SELECT_FOR_UNLOAD so that the SELECT privilege continues to be checked when the UNLOAD utility runs. For more information, see AUTH_COMPATIBILITY in macro DSN6SPRM (Db2 Installation and Migration).

Db2 command release incompatibilities

When you migrate from Db2 11 to Db2 12, or apply maintenance in Db2 12, be aware of the command release incompatibilities.

GUPI

Plan for the following changes in Db2 12 that might affect your migration.

The following incompatible changes apply at any Db2 12 function level, including when you first migrate to Db2 12. For incompatible changes that might impact your Db2 12 environment when you activate function levels 501 and higher, see "Incompatible changes summary for function levels 501 and higher" on page 119.

Release incompatibilities that were changed or added since the first edition of this Db2 12 publication are indicated by a vertical bar in the left margin. In other areas of this publication, a vertical bar in the margin indicates a change or addition that has occurred since the Db2 11 release of this publication.

Change to DISPLAY UTILITY output

The output for the DISPLAY UTILITY command now includes message DSNU285I, regardless of whether the SWITCHTIME parameter was specified in the REORG INDEX or REORG TABLESPACE utility control statement. In Db2 11, message DSNU285I was displayed only if SWITCHTIME was specified in the REORG INDEX or REORG TABLESPACE utility control statement.

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

Actions to take

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

GUPI

Related information
DSNU100I (Db2 Messages)

Changed REBIND command result for SWITCH option with invalid package copies

If you issue a REBIND PACKAGE or REBIND TRIGGER PACKAGE command with the SWITCH option and you specify an invalid package, the command now fails.

In previous releases, the only productive reason to switch to an invalid package copy was as the first action of a two-step process. The second step was to rebind the packages with the APREUSE option to reuse the access path information from the invalid package as input for access path reuse.

Actions to take

If you previously used the two-step process to reuse access paths from invalid copies, modify your process to use a single step. Rebind with the APREUSE option and specify PREVIOUS or ORIGINAL for the APREUSESOURCE bind option to specify whether the previous or original package copy as input for access path reuse.

Related tasks
Reverting to saved access paths for static SQL statements (Db2 Performance)
Related reference
APREUSESOURCE bind option (Db2 Commands)
APREUSE bind option (Db2 Commands)

Related information
DSNT330I (Db2 Messages)
-4750 (Db2 Codes)

Change to START RLIMIT command result

Explanation

Messages DSNT731I and DSNT732I might be issued for START RLIMIT commands if resource limit tables are detected with deprecated or unsupported formats.

Starting in Db2 12, DSNRLST*xx* table formats and related index formats from before DB2 version 8 format are not supported. When Db2 detects DSNRLST*xx* tables with unsupported formats, it issues message DSNT731I.

DSNRLMT*xx* table formats and related index formats from before Db2 11 are deprecated. In Db2 12, if Db2 detects DSNRLMT*xx* tables in a deprecated format, it issues message DSNT732I, processing for the START RLIMIT command continues, and the resource limit facility starts using the deprecated objects.

Dynamic SQL applications that are specified in the obsolete resource limit tables can run without the limits.

Actions to take

Update existing resource limit tables to use the supported formats. The DSNTIJSG installation job contains statements that create the database, table space, tables, and indexes for the resource limit facility. You can tailor those statements to create or update the format of resource limit tables.

Related tasks
Run premigration queries (DSNTIJPM) (Db2 Installation and Migration)
Related reference
-START RLIMIT (Db2) (Db2 Commands)
Resource limit facility tables (Db2 Performance)
Job DSNTIJSG (Db2 Installation and Migration)

Security release incompatibilities

When you migrate to or apply maintenance in Db2 12, be aware of security release incompatibilities that might affect your Db2 environment.

The following incompatible changes apply at any Db2 12 function level, including when you first migrate to Db2 12. For incompatible changes that might impact your Db2 12 environment when you activate function levels 501 and higher, see "Incompatible changes summary for function levels 501 and higher" on page 119.

Release incompatibilities that were changed or added since the first edition of this Db2 12 publication are indicated by a vertical bar in the left margin. In other areas of this publication, a vertical bar in the margin indicates a change or addition that has occurred since the Db2 11 release of this publication.

Change to RACF long object resource names

For UPDATE and REFERENCES operations on tables, table-qualifiers and table-names are each truncated at 100 characters. The column-name is truncated at 30 characters.

When you use long object names for UPDATE and REFERENCES operations, along with discrete RACF profiles, truncation can cause unintended results. If truncation occurs, a single discrete profile might inadvertently protect multiple resources if the resources are similarly named, the first 100 characters of the schema names are identical, and the qualified object names, such as the table name, subsystem name, or privilege name, are also identical.

Actions to take

Review the RACF profiles for long table names for UPDATE and REFERENCES operations, and update those profiles, as needed.

Changes to the authorization check of the UNLOAD utility

After the activation of function level 500 or higher, Db2 checks for the new UNLOAD privilege, instead of the SELECT privilege, for executing the UNLOAD utility and UNLOAD jobs might fail due to authorization errors.

Db2 checks the new UNLOAD privilege for authorization of the UNLOAD utility. You must use a privilege set that includes one of the following to execute the UNLOAD utility:

- Ownership of the tables
- UNLOAD privilege on the tables
- SELECT privilege on the tables (if the AUTH_COMPTIBILITY subsystem parameter is set to SELECT_FOR_UNLOAD)
- DBADM authority for the database. If the object on which the utility operates is in an implicitly created database, DBADM authority on DSNDB04 or the implicitly created database is sufficient.
- DATAACCESS authority
- SYSADM authority
- SYSCTRL authority (catalog tables only)
- SQLADM authority (catalog tables only)
- System DBADM authority (catalog tables only)
- ACCESSCTRL authority (catalog tables only)
- SECADM authority (catalog tables only)

When an UNLOAD job is run by a user with only the SELECT privilege on the target table, the UNLOAD job fails with message DSNU1253I.

Actions to take

Take any of the following actions:

- Before migration to Db2 12, run job DSNTIJPM, and follow the steps in premigration report 19 to determine which authorization IDs might need the UNLOAD privilege.
- In Db2 11 or Db2 12, start a trace for IFCID 404 to identify the users who need the UNLOAD privilege. Then, in Db2 12 at function level 100, grant the UNLOAD privilege to those users.
- As a temporary solution, set the AUTH_COMPATIBILITY subsystem parameter to SELECT_FOR_UNLOAD so that the SELECT privilege continues to be checked when the UNLOAD utility runs. For more information, see AUTH_COMPATIBILITY in macro DSN6SPRM (Db2 Installation and Migration).

Storage release incompatibilities in Db2 12

When you migrate to or apply maintenance in Db2 12, be aware of and plan for storage release incompatibilities that might affect your Db2 environment.

The following incompatible changes apply at any Db2 12 function level, including when you first migrate to Db2 12. For incompatible changes that might impact your Db2 12 environment when you activate function levels 501 and higher, see "Incompatible changes summary for function levels 501 and higher" on page 119.

Release incompatibilities that were changed or added since the first edition of this Db2 12 publication are indicated by a vertical bar in the left margin. In other areas of this publication, a vertical bar in the margin indicates a change or addition that has occurred since the Db2 11 release of this publication.

Changed behavior for PGSTEAL(NONE) buffer pools

PGSTEAL(NONE) buffer pools are managed differently in Db2 12.

Explanation

Db2 12 introduces the following changes to management of PGSTEAL(NONE) buffer pools:

- In Db2 12, Db2 keeps all pages for open objects that are assigned to PGSTEAL(NONE) buffer pools in memory. When the buffer pool is allocated, Db2 creates an overflow area within the VPSIZE of the buffer pool. The overflow area is used if the buffer pool is not large enough to contain all pages for all open assigned objects. Page stealing can also occur in the overflow area. LRU chains are maintained for the overflow area, but buffer stealing in the overflow area uses the FIFO page-stealing algorithm.
- In Db2 12, if you specify PGSTEAL(NONE) with FRAMESIZE(2G) in an ALTER BUFFERPOOL command, when Db2 allocates the buffer pools, it temporarily changes the page-stealing algorithm to PGSTEAL(LRU) and issues message DSNB549I.

  In Db2 11, FRAMESIZE(2G) was allowed with PGSTEAL(NONE), but this combination might waste real storage and lead to storage shortages.

Possible impact to your Db2 environment

If a PGSTEAL(NONE) buffer pool is not large enough to contain all open assigned objects plus the overflow area, the overflow area is used. Pages that are allocated to the overflow area are not reallocated in the main part of the buffer pool until the data set is closed. Page stealing can occur in the overflow area. LRU chains are maintained for the overflow area, but buffer stealing in the overflow area uses the FIFO page-stealing algorithm.

Actions to take

Verify that existing buffer pools that are defined with the PGSTEAL(NONE) option meet the following criteria:

- Assigned objects are frequently accessed, and not likely to grow quickly. For example, you can use the following real-time statistics values to identify candidates:
  - GETPAGES
  - NACTIVE
  - REORGINSERTS
- The VPSIZE of the buffer pool is large enough to contain all pages for all assigned objects, and the overflow area. The size of the overflow is approximately 10 percent of the VPSIZE value for the buffer pool. The minimum size of the overflow area is 50 buffers, and the maximum size is 6400 buffers.

Alter buffer pools that do not meet the criteria to use the PGSTEAL(FIFO) or PGSTEAL(LRU) option.

Related tasks
Choosing a page-stealing algorithm (Db2 Performance)
Related reference
-ALTER BUFFERPOOL (Db2) (Db2 Commands)
Related information
DSNB416I (Db2 Messages)
DSNB549I (Db2 Messages)

Change to storage handling for the dynamic statement cache

Db2 12 changes the way in which storage in the EDM pool is allocated for cached dynamic statements.

Explanation

In Db2 12, EDMSTMTC specifies an upper limit on the amount of storage in the EDM pool that is used for cached dynamic statements. EDMSTMTC does not specify the initial amount of storage that is allocated.

Storage is allocated only as it is needed. If adding of statements to the dynamic statement cache raises the amount of storage to a value that is greater than EDMSTMTC, the Db2 database manager removes statements that are not in use from the cache to lower the amount of storage to the EDMSTMTC value.

In previous releases, EDMSTMTC specified the initial amount of storage in the EDM pool that was allocated for cached dynamic statements. The initial EDMSTMTC value also established the minimum amount of storage that could be used for cached dynamic statements until the Db2 subsystem was restarted.

Possible impact to your Db2 environment
More real storage might be needed.

Actions to take

If storage shortages occur, increase the amount of real storage that is available.

Related tasks
Calculating the EDM statement cache hit ratio (Db2 Performance)
Related reference
EDM STATEMENT CACHE field (EDMSTMTC subsystem parameter) (Db2 Installation and Migration)

Increased EDM storage usage for DBDs
Starting with Db2 12 function level V12R1M100, you might need more EDM pool storage for DBDs than you needed in Db2 11.

Explanation

Immediately after migration to Db2 12 function level V12R1M100, for tables and table spaces that were created prior to Db2 12, more space is needed for storing their internal data structures in the EDM pool. However, after new function activation (Db2 12 function level V12R1M500 or later), when a data definition statement such as an ALTER is performed on one of those tables or table spaces, the storage usage for their internal structures is reduced again.

Possible impact to your Db2 environment
More EDM pool storage for DBDs might be needed after migration to Db2 12 function level V12R1M100, and before a data definition change to a table or table space in Db2 12 function level V12R1M500 or later.

Actions to take

Issue the following query after you migrate to Db2 12 function level V12R1M100. Then update the EDMDBDC subsystem parameter with a value that is at least twice the query result.

```
WITH DBD_LEN_TABLE (DBD_LEN) AS
  (SELECT SUM(LENGTH(A.DBD_DATA)) AS DBD_LEN
    FROM SYSIBM.DBDR A,SYSIBM.SYSDATABASE B
    WHERE  A.DBID = B.DBID
    GROUP BY A.DBID
    ORDER BY DBD_LEN DESC)
 SELECT SUM(DBD_LEN)/1024 AS EDMDBDC_IN_KBYTES
  FROM DBD_LEN_TABLE
  WITH UR;
```

Related reference
EDM DBD CACHE field (EDMDBDC subsystem parameter) (Db2 Installation and Migration)

Subsystem parameter release incompatibilities in Db2 12

When you migrate to or apply maintenance in Db2 12, be aware of and plan for subsystem parameter release incompatibilities that might affect your Db2 environment.

The following incompatible changes apply at any Db2 12 function level, including when you first migrate to Db2 12. For incompatible changes that might impact your Db2 12 environment when you activate function levels 501 and higher, see "Incompatible changes summary for function levels 501 and higher" on page 119.

Related tasks
Updating subsystem parameter and application default values (Db2 Installation and Migration)
Related reference
Subsystem parameter changes in Db2 12
You can use this information to plan for subsystem parameter changes in Db2 12.

Removal of the COMPRESS_SPT01 subsystem parameter

Removal of the COMPRESS_SPT01 parameter will cause Db2 12 to operate as if that parameter were set to YES, a switch from the current default behavior of NO. In data sharing, all members of the group need to use the same setting for COMPRESS_SPT01. If your SPT01 table space is not compressed, Db2 will always compress it the next time you reorganize it.

Actions to take

Use an IFCID 106 trace record formatter such as the SYSPROC.ADMIN_INFO_SYSPARM stored procedure to check the current setting of COMPRESS_SPT01. If the setting is YES then the change to remove COMPRESS_SPT01 is compatible with your environment and no further action is required. Otherwise, proceed as follows:

- If you currently run Db2 12: before applying the fixing PTF, modify your subsystem parameter module to specify COMPRESS_SPT01=YES. In data sharing, ensure that all members of the group use COMPRESS_SPT01=YES before applying the fixing PTF to any member of the group.

- If you currently run Db2 11: before migrating to Db2 12, modify your subsystem parameter module to specify COMPRESS_SPT01=YES. In data sharing, ensure that all members of the group use COMPRESS_SPT01=YES before applying the fixing PTF to any member of the group.

Removal of the SPT01_INLINE_LENGTH subsystem parameter

Removal of SPT01_INLINE_LENGTH causes Db2 12 to operate as if that parameter were set to 32138, the maximum and current default setting. In data sharing, all members of the group need to use the same setting for SPT01_INLINE_LENGTH.

If your SPT01 table space has an inline LOB length of less than 32138, Db2 will always place it in AREO* the next time the -SET SYSPARM command is issued.

Actions to take

Use an IFCID 106 trace record formatter such as the SYSPROC.ADMIN_INFO_SYSPARM stored procedure to check the current setting of SPT01_INLINE_LENGTH. If the setting is 32138, the change to remove SPT01_INLINE_LENGTH is compatible with your environment and no further action is required. Otherwise, proceed as follows:

- If you currently run Db2 12: before applying the fixing PTF, modify your subsystem parameter module to specify SPT01_INLINE_LENGTH=32138. In data sharing environments, ensure that all members of the group use SPT01_INLINE_LENGTH=32138 before applying the fixing PTF to any member of the group.

- If you currently run Db2 11: before migrating to Db2 12, modify your subsystem parameter module to specify SPT01_INLINE_LENGTH=32138. In data sharing environments, ensure that all members of the group use SPT01_INLINE_LENGTH=32138 before migrating any member of the group.

Removal of the OBJECT_CREATE_FORMAT subsystem parameter

Removal of OBJECT_CREATE_FORMAT subsystem parameter causes Db2 12 to operate as if that parameter were set to EXTENDED, the current default setting. In data sharing, all members of the group need to use the same setting for OBJECT_CREATE_FORMAT. With this setting, Db2 always creates new table spaces and indexes to use the extended 10-byte log record format.

Actions to take

Use an IFCID 106 trace record formatter such as the SYSPROC.ADMIN_INFO_SYSPARM stored procedure to check the current setting of OBJECT_CREATE_FORMAT. If the setting is EXTENDED, the change to remove OBJECT_CREATE_FORMAT is compatible with your environment and no further action is required. Otherwise, proceed as follows:

- If you currently run Db2 12: before applying the fixing PTF, modify your subsystem parameter module to specify OBJECT_CREATE_FORMAT=EXTENDED. In data sharing environments, ensure that all members of the group use OBJECT_CREATE_FORMAT=EXTENDED before applying the fixing PTF to any member of the group.

- If you currently run Db2 11: before migrating to Db2 12, modify your subsystem parameter module to specify OBJECT_CREATE_FORMAT=EXTENDED. In data sharing environments, ensure that all members of the group use OBJECT_CREATE_FORMAT=EXTENDED before migrating any member of the group.

Related concepts
The extended 10-byte RBA and LRSN in Db2 12
Beginning in Db2 11, Db2 uses the extended format 10-byte RBA and LRSN values, and conversion of the BSDS to the extended format is required at migration to Db2 12.

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

Removal of the UTILITY_OBJECT_CONVERSION subsystem parameter

With APAR PH26317, removal of UTILITY_OBJECT_CONVERSION causes Db2 12 to operate as if that parameter were set to NOBASIC, a switch from the current default setting of NONE. In data sharing, all members of the group need to use the same setting for UTILITY_OBJECT_CONVERSION. With this setting, Db2 utilities that accept the RBALRSN_CONVERSION option always convert existing table spaces and indexes from the traditional 6-byte page (BASIC) format to the 10-byte page (EXTENDED) format.

Important: When the NOBASIC setting of the UTILITY_OBJECT_CONVERSION subsystem parameter is in effect, any LOAD, REORG, and REBUILD INDEX utility job that specifies RBALRSN_CONVERSION BASIC always fails. Similarly, if the target object is still in the 6-byte page format, these utility jobs also fail if they specify RBALRSN_CONVERSION NONE. Plan to remove the RBALRSN_CONVERSION option from these utility jobs before applying the PTF for PH26317.

Actions to take

Use an IFCID 106 trace record formatter such as the SYSPROC.ADMIN_INFO_SYSPARM stored procedure to check the current setting of UTILITY_OBJECT_CONVERSION. If the setting is NOBASIC then the change to remove UTILITY_OBJECT_CONVERSION is compatible with your environment and no further action is required. Otherwise, proceed as follows:

- **If your LOAD, REORG, and REBUILD INDEX utility jobs specify RBALRSN_CONVERSION BASIC or RBALRSN_CONVERSION BASIC:** Remove the RBALRSN_CONVERSION keyword from the utility jobs. Otherwise, these jobs are expected to begin failing with message DSNU1145I.

- **If you currently run Db2 12:** before applying the fixing PTF, modify your subsystem parameter module to specify UTILITY_OBJECT_CONVERSION=NOBASIC. In data sharing environments, ensure that all members of the group use UTILITY_OBJECT_CONVERSION=NOBASIC before applying the fixing PTF to any member of the group.

Any existing table space or index having the traditional 6-byte page (BASIC) format is converted to the 10-byte page (EXTENDED) format the next time it is processed by a Db2 utility that accepts the RBALRSN_CONVERSION option. Also, existing table spaces and indexes having the 10-byte page (EXTENDED) format can no longer be converted to the deprecated 6-byte page (BASIC) format.

- **If you currently run Db2 11:** before migrating to Db2 12, modify your subsystem parameter module to specify UTILITY_OBJECT_CONVERSION=NOBASIC. In data sharing environments, ensure that all members of the group use UTILITY_OBJECT_CONVERSION=NOBASIC before migrating any member of the group.

  Any existing table space or index in the deprecated 6-byte page (BASIC) format is converted to the 10-byte page (EXTENDED) format the next time it is processed by a Db2 utility that accepts the RBALRSN_CONVERSION option. Also, existing table spaces and indexes having the 10-byte page (EXTENDED) format can no longer be converted to the deprecated 6-byte page (BASIC) format.

Important: After migration to Db2 12, the first time that REORG runs on an XML table that supports multiple versions, Db2 alters the data type of the START_TS and END_TS columns in the XML table from BINARY(8) to BINARY(10). That column alteration causes invalidation of dependent packages. As a result of package invalidation, Db2 performs an automatic rebind on application packages that access the XML table. If the first run of the REORG utility on an XML table in Db2 12 occurs at the same time that applications with invalidated packages are running, resource contention can occur, and the REORG or the applications might time out.

Related concepts
The extended 10-byte RBA and LRSN in Db2 12
Beginning in Db2 11, Db2 uses the extended format 10-byte RBA and LRSN values, and conversion of the BSDS to the extended format is required at migration to Db2 12.

Related tasks
Converting page sets to the 10-byte RBA or LRSN format (Db2 Administration Guide)
Related information
PH26317

Changes to REORG TABLESPACE PART SORTNPSI YES behavior

Starting in Db2 12 at function level 100, when you specify the REORG TABLESPACE utility option SORTNPSI YES or set the REORG_PART_SORT_NPSI subsystem parameter to YES, REORG sorts the keys for all non-partitioned secondary indexes (NPSIs). Previously, when you specified these options, REORG sorted NPSIs only when certain internal thresholds were met. Therefore, some REORG SORTNPSI YES jobs that previously did not qualify for a sort operation will include a sort operation in Db2 12. Sorts can impact performance and might increase the elapsed time of those utility jobs.

Actions to take

If the extra sort operations cause a performance impact, set the subsystem parameter REORG_PART_SORT_NPSI to AUTO, or specify SORTNPSI AUTO in the REORG TABLESPACE utility statement.

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

Maximum number of user-defined external scalar functions running in a Db2 thread is no longer unlimited (APAR PH44833)

Starting at application compatibility level V12R1M100 (if the PTF for APAR PH44833 is applied), Db2 12 introduces the MAX_UDF subsystem parameter. MAX_UDF 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.

Actions to take

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.

Related reference
MAX UDFS field (MAX_UDF subsystem parameter) (Db2 Installation and Migration)
Related information
00E70082 (Db2 Codes)

BSDS record format conversion is required

The BSDS must be converted to use the 10-byte format RBA and LRSN records before Db2 12 can start.

At migration to Db2 12, you cannot start Db2 12 until the BSDS is converted to use the 10-byte RBA and LRSN formats. You can convert the BSDS before or during the Db2 12 migration process.

Before you can convert the BSDS, the Db2 subsystem or data sharing member must be started in Db2 11 new-function mode.

⚠️ **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).

In a data sharing environment, page sets contain LRSNs instead of RBAs. After the BSDS has been converted to allow for 10-byte LRSNs, an LRSN that is greater than 6 bytes does not result in failed updates with reason code 00C2026D. It is recommended that after the BSDS has been converted, the catalog and directory table spaces are converted before any other objects. Other objects need to be converted to extended format before the LRSN value reaches the 6-byte LRSN limit.

For instructions, see Convert BSDS records to the extended 10-byte format: DSNTIJCB (Db2 Installation and Migration).

Related concepts
The extended 10-byte RBA and LRSN in Db2 12
Beginning in Db2 11, Db2 uses the extended format 10-byte RBA and LRSN values, and conversion of the BSDS to the extended format is required at migration to Db2 12.

Related tasks
Migration step 18: Start Db2 12 (Db2 Installation and Migration)
Related reference
DSNJCNVT (Db2 Utilities)

Storage changes in Db2 12

Db2 12 introduces changes to the storage configuration for a Db2 for z/OS installation.

database services address space (*ssnm*DBM1) MEMLIMT increased to 19T

In job DSNTIJMV, which defines Db2 to the z/OS operating system, the default MEMLIMIT setting for the *ssid*database services address space (*ssnm*DBM1) is increased from 4 TB to 19 TB, to support the new 16 TB limit for all buffer pool storage.

MEMLIMIT specifies the limit on the total size of usable virtual storage above the bar in a single address space.

New requirements for 32KB work files

For migration to Db2 12, you might need to increase your 32KB work file allocations for work file data sets and work file buffer pools. If you did not define 32KB work files in previous Db2 releases, you might need to define 32KB work files in Db2 12. To implement improvements in sort processing and performance, Db2 12 uses more 32KB work files than previous Db2 releases. For sorting a very large number of records, Db2 12 might require 32KB work files, even though previous releases did not use them.

When Db2 12 uses 32KB work files for sorting a very large number of records, and there is an insufficient amount of 32KB work file space, Db2 12 cannot use work files of a smaller size. The operation that requires the sort fails with an error that indicates that there is not enough work file space.

Changed behavior for PGSTEAL(NONE) buffer pools

Db2 12 introduces the following changes to management of PGSTEAL(NONE) buffer pools:

- In Db2 12, Db2 keeps all pages for open objects that are assigned to PGSTEAL(NONE) buffer pools in memory. When the buffer pool is allocated, Db2 creates an overflow area within the VPSIZE of the buffer pool. The overflow area is used if the buffer pool is not large enough to contain all pages for all open assigned objects. Page stealing can also occur in the overflow area. LRU chains are maintained for the overflow area, but buffer stealing in the overflow area uses the FIFO page-stealing algorithm.
- In Db2 12, if you specify PGSTEAL(NONE) with FRAMESIZE(2G) in an ALTER BUFFERPOOL command, when Db2 allocates the buffer pools, it temporarily changes the page-stealing algorithm to PGSTEAL(LRU) and issues message DSNB549I.

  In Db2 11, FRAMESIZE(2G) was allowed with PGSTEAL(NONE), but this combination might waste real storage and lead to storage shortages.

For more information about this change, see "Storage release incompatibilities in Db2 12" on page 141.

Related concepts
Buffer pools (Introduction to Db2 for z/OS)
Work file database storage requirements (Db2 Installation and Migration)
Related tasks
Installation step 2: Define Db2 to IRLM and MVS: DSNTIJMV (Db2 Installation and Migration)
Migration step 13: Define Db2 12 to IRLM and MVS: DSNTIJMV (Db2 Installation and Migration)

Command changes in Db2 12

Db2 12 introduces new commands and changes to existing commands.

⟨GUPI⟩

Important: Db2 12 introduces continuous delivery of new capabilities and enhancements in function levels. Most new capabilities become available only after activation of the Db2 12 function level that introduces them, or when applications run with the corresponding application compatibility level. For more information, see Chapter 12, "Activating Db2 12 function levels," on page 281.

Command changes in function level 501 and higher

The following changes to commands take effect when you activate function level 501 or higher in Db2 12.

Function level	Command	Change introduced	Incompatible change?
FL 502	-DISPLAY ARCHIVE (Db2) (Db2 Commands)	The DISPLAY ARCHIVE command displays the current key label information for active logs and archive logs.	No
FL 502	-DISPLAY GROUP (Db2) (Db2 Commands)	The DISPLAY GROUP command output now provides the current ENCRYPTION_KEYLABEL system parameter value.	No
FL 502	-DISPLAY LOG (Db2) (Db2 Commands)	The DISPLAY LOG command now displays the current key label information for active logs and archive logs.	No

Command changes in function level 500

The following changes to commands take effect when you activate function level 500 in Db2 12.

Command	Change introduced	Incompatible change?
-ACCESS DATABASE (Db2) (Db2 Commands)	If the externalization of real-time statistics (RTS) fails, Db2 now issues message DSNT538I.	Yes
-ALTER BUFFERPOOL (Db2) (Db2 Commands)	The meaning of the existing PGSTEAL(NONE) option is changed. As in previous versions, Db2 pre-loads the data for assigned objects when they are opened and keeps the data resident until the objects are closed. However, in Db2 12, Db2 creates an overflow area within the VPSIZE of the buffer pool. If all open assigned objects do not fit into the main part of the buffer pool, pages are allocated to the overflow area. Page stealing is possible in the overflow area. After pages are allocated in the overflow area, you must close the object to move the pages back to the overflow area.	No
-ALTER GROUPBUFFERPOOL (Db2) (Db2 Commands)	The maximum value that can be specified for the RATIO keyword is now increased to 1024.	No
BIND PACKAGE subcommand (DSN) (Db2 Commands)	The following new option is added: CONCENTRATESTMT.	No
BIND QUERY (DSN) (Db2 Commands)	Support is added for processing the following SQL statement types: • INSERT • UPDATE WHERE CURRENT OF • DELETE WHERE CURRENT OF	No
-DISPLAY DYNQUERYCAPTURE (Db2) (Db2 Commands)	New command.	No
-DISPLAY GROUP (Db2) (Db2 Commands)	The DSN7100I message output is modified to indicate information about the code, catalog, and function levels of the Db2 subsystem or data sharing group.	No

Command	Change introduced	Incompatible change?
-DISPLAY ML (Db2) (Db2 Commands)	New command. "1" on page 151	No.
-DISPLAY STATS (Db2) (Db2 Commands)	New command. New options: • INDEXTRAVERSECOUNT is used to display the index traverse counts for a specific index, or for a specified number of indexes with the highest traverse counts, in descending order. Output is returned by message DSNT830I (Db2 Messages). • LOGREADERTASKS and LOGREADERTASKS SCOPE(GROUP) options are used to display statistics about any log reading tasks that are currently running. Output is returned by message DSNT788I.	No
-DISPLAY UTILITY (Db2) (Db2 Commands)	The output now includes message DSNU285I.	Yes
FREE STABILIZED DYNAMIC QUERY (DSN) (Db2 Commands)	New command.	No
FREE PACKAGE (DSN) (Db2 Commands)	If you specify INACTIVE, PREVIOUS, ORIGINAL for the PLANMGMTSCOPE option, FREE PACKAGE processing can now complete while applications that use the package are running. New values are added for an existing option: PLANMGMTSCOPE(PREVIOUS|ORIGINAL). A new option is added: INVALIDONLY A new value is added to an existing option: PLANMGMTSCOPE(PHASEOUT).	No
FREE SERVICE (DSN) (Db2 Commands)	New command. SERVICE now supports specifying *location-name* and *version-id*.	No
-MODIFY DDF (Db2) (Db2 Commands)	After the PTF for APAR PH08188 is applied, PORT and SECPORT can now have the name value. A new option is added: SESSIDLE.	No
REBIND PACKAGE subcommand (DSN) (Db2 Commands)	This command can be used to rebind a package for an advanced trigger. The following new options are added: • APREUSESOURCE • CONCENTRATESTMT If the SWTICH option is specified with an invalid package, the command now fails.	Yes
REBIND TRIGGER PACKAGE (DSN) (Db2 Commands)	This command can be used to rebind a package for a basic trigger, including any trigger that was	Yes

Command	Change introduced	Incompatible change?
	created prior to the activation of function level 500 or higher. A new option is added: APREUSESOURCE. If the SWTICH option is specified with an invalid package, the command now fails.	
-START DB2 (Db2) (Db2 Commands)	You cannot start Db2 at any code level that is lower than the catalog level or the highest activated function level.	No
-START PROFILE (Db2) (Db2 Commands)	If the PROFILE_AUTOSTART subsystem parameter is set to YES, this command is now issued automatically when Db2 starts.	No
-START ML (Db2) (Db2 Commands)	New command. [1] on page 151 New options added: *, SQLOPT, and SA_DCC.	No
-START RLIMIT (Db2) (Db2 Commands)	Certain previously supported resource limit table formats are deprecated or unsupported in Db2 12. Resource limits are not activated for tables with unsupported formats.	No
-STOP ML (Db2) (Db2 Commands)	New command. [1] on page 151 New options added: *, SQLOPT, and SA_DCC.	No
-TERM UTILITY (Db2) (Db2 Commands)	The TERM UTILITY command can be used for the MODIFY RECOVERY utility.	No
-START CDDS (Db2) (Db2 Commands)	New command.	No
-STOP CDDS (Db2) (Db2 Commands)	New command.	No
-START DYNQUERYCAPTURE (Db2) (Db2 Commands)	New command.	No
-STOP DYNQUERYCAPTURE (Db2) (Db2 Commands)	New command.	No

Notes:

1. With IBM Db2 AI for z/OS Version 1.1.0.1 or later and APAR PH05323.

Command changes in function level 100

The following changes to commands take effect when you migrate to function level 100 in Db2 12 from Db2 11.

Command	Change introduced	Incompatible change?
-ACTIVATE (Db2) (Db2 Commands)	New command.	No
BIND SERVICE (DSN) (Db2 Commands)	New command.	No

Command	Change introduced	Incompatible change?
	The following options are added: • ACCELERATIONWAITFORDATA • COPY • COPYVER • OPTIONS • VERSION	
-DISPLAY ACCEL (Db2) (Db2 Commands)	If the DETAIL keyword is specified for multiple accelerators, Db2 12 issues a separate DSNX830I message is issued for each accelerator.	No
-DISPLAY BLOCKERS (Db2) (Db2 Commands)	New command.	No
-DISPLAY LOCATION (Db2) (Db2 Commands)	New option: INCOMPLT	No
-DISPLAY RESTSVC (Db2) (Db2 Commands)	New command.	No
FREE SERVICE (DSN) (Db2 Commands)	New command.	No
-MODIFY DDF (Db2) (Db2 Commands)	New option: ACCTG	No
REBIND PACKAGE subcommand (DSN) (Db2 Commands)	The following options are added: • RESTSERVICEDEFAULT	No
-START RLIMIT (Db2) (Db2 Commands)	Resource limits that apply to existing threads can now be modified by issuing the START RLIMIT command. A new option is added: SCOPE.	No
-START RESTSVC (Db2) (Db2 Commands)	New command.	No
-STOP RESTSVC (Db2) (Db2 Commands)	New command.	No

GUPI

SQL changes in Db2 12

You can use this information to plan for SQL changes in Db2 12.

Important: Db2 12 introduces continuous delivery of new capabilities and enhancements in function levels. Most new capabilities become available only after activation of the Db2 12 function level that introduces them, or when applications run with the corresponding application compatibility level. For more information, see Chapter 12, "Activating Db2 12 function levels," on page 281.

Important: Applications must run with V12R1M500 application compatibility or higher to use most of these new SQL capabilities. For more information, see Application compatibility levels in Db2 12 (Db2 Application programming and SQL).

SQL statement changes in function level 501 and higher

APPLCOM PAT level	SQL element	Change introduced	Incompatible change?
V12R1M509	ALTER TABLESPACE	The following new clauses are added: • COMPRESS YES FIXEDLENGTH • COMPRESS YES HUFFMAN	No
V12R1M509	CREATE TABLE	The following new clauses are added: • COMPRESS YES FIXEDLENGTH • COMPRESS YES HUFFMAN IN ACCELERATOR can now specify an alias that represents multiple accelerators.	No
V12R1M509	CREATE TABLESPACE	The following new clauses are added: • COMPRESS YES FIXEDLENGTH • COMPRESS YES HUFFMAN	No
V12R1M509	DELETE	A restriction against specifying FOR PORTION OF BUSINESS_TIME if the target table is a parent in a temporal referential constraint is removed.	No
V12R1M508	ALTER TABLESPACE	New clause: MOVE TABLE.	No
V12R1M507	CREATE PROCEDURE (external)	The following new clauses are added: • OR REPLACE • SPECIFIC	No
V12R1M507	CREATE PROCEDURE (SQL native)	The following new clauses are added: • OR REPLACE • SPECIFIC	No
V12R1M506	DROP	Changed clause: TABLE.	No
V12R1M505	ALTER TABLE	The following clauses are now supported for DECFLOAT columns: PRIMARY KEY and UNIQUE.	No
V12R1M505	CREATE INDEX	*column-name* can now specify DECFLOAT columns.	No
V12R1M505	CREATE TABLE	The following clauses now support DECFLOAT columns: PRIMARY KEY and UNIQUE.	No
V12R1M505	CREATE TRIGGER (basic)	Changed clause: WHEN *search-condition*.	No
V12R1M505	CREATE TRIGGER (advanced)	Changed clause: WHEN *search-condition*.	No
V12R1M503	ALTER TABLE	New clause: ON DELETE ADD EXTRA ROW.	No

APPLCOM PAT level	SQL element	Change introduced	Incompatible change?
V12R1M502	ALTER STOGROUP (Db2 SQL)	New clause: KEY LABEL.	No
V12R1M502	ALTER TABLE	New clause: KEY LABEL.	No
V12R1M502	CREATE STOGROUP	New clause: KEY LABEL.	No
V12R1M502	CREATE TABLE	New clause: KEY LABEL.	No

SQL statement changes in function level 500

The following SQL changes take effect in Db2 12 for applications that run at application compatibility V12R1M500 or higher.

SQL element	Change introduced	Incompatible change?
ALTER FUNCTION (compiled SQL scalar)	New clause: CONCENTRATE STATEMENTS.	No
ALTER INDEX	A new clauses are added: DSSIZE. The following clauses are changed: • ADD COLUMN *column-name* (can now specify DECFLOAT columns) • COMPRESS Altering to use index compression for indexes in universal table spaces is now a pending change that places the index in advisory REORG-pending (AREOR) status	Yes
ALTER PROCEDURE (SQL native)	New clause: CONCENTRATE STATEMENTS.	No
ALTER TABLE	The following new clauses are added: • CCSID • EXCLUSIVE and INCLUSIVE clauses of the PERIOD BUSINESS_TIME clause • PERIOD BUSINESS_TIME clause for constraints The following clauses are changed: • ADD PERIOD FOR • ADD PARTITION	No
ALTER TABLESPACE	The following new clauses are added: • PAGENUM • INSERT ALGORITHM The following clauses are changed: • COMPRESS	No

SQL element	Change introduced	Incompatible change?
	• DSSIZE	
ALTER TRIGGER (advanced) (Db2 SQL)	New statement.	No
ALTER TRIGGER (basic)	Equivalent to ALTER TRIGGER in prior releases.	No
COMMENT (Db2 SQL)	Changed clause: TRIGGER *trigger-name* VERSION *trigger-version-id*.	No
CREATE FUNCTION (compiled SQL scalar)	The following new clauses are added: • CONCENTRATE STATEMENTS • WRAPPED	No
CREATE FUNCTION (inline SQL scalar)	New clause: WRAPPED.	No
CREATE FUNCTION (SQL table)	New clause: WRAPPED.	No
CREATE INDEX	New clause: DSSIZE. CREATE INDEX statements that omit the USING clause at the table space or index level now fail with SQLCODE -204, if the storage group specified when the containing database was created does not exist.	Yes
CREATE PROCEDURE (SQL native)	The following new clauses are added: • CONCENTRATE STATEMENTS • WRAPPED	No
CREATE TABLE	The following new clauses are added: • CCSID on a CHAR, GRAPHIC, CLOB, or DBCLOB column • EXCLUSIVE and INCLUSIVE clauses of the PERIOD BUSINESS_TIME clause • PAGENUM • PERIOD BUSINESS_TIME for constraints The following clauses are changed: • DSSIZE • PERIOD FOR	No
CREATE TABLESPACE	The following new clauses are added: • PAGENUM • INSERT ALGORITHM The following clauses are changed: • COMPRESS • DSSIZE • NUMPARTS	Yes

SQL element	Change introduced	Incompatible change?
	CREATE TABLESPACE statements that omit the USING clause at the table space or index level now fail with SQLCODE -204, if the storage group specified when the containing database was created does not exist.	
CREATE TRIGGER (basic)	New clause: WRAPPED.	No
CREATE TRIGGER (advanced)	New statement.	No
CREATE VARIABLE	Changed clause: *data-type*.	No
DELETE	The following new clauses are added: • BETWEEN *value-1* AND *value-2* clause of the *period-clause* • FETCH FIRST *n* ROWS ONLY	No
EXECUTE	Changed clause: USING.	No
EXECUTE IMMEDIATE (Db2 SQL)	Changed clause: *variable*.	No
EXPLAIN	New clause: STABILIZED DYNAMIC QUERY STMTID. When Db2 processes the SQL statement EXPLAIN PACKAGE or EXPLAIN STABILIZED DYNAMIC QUERY, the HINT_USED column in the PLAN_TABLE is populated with EXPLAIN PACKAGE: *copy*.	Yes
FETCH	New clause: *target-variable*.	No
fullselect (Db2 SQL)	New clause: offset-clause (Db2 SQL). The following clauses are changed: • fetch-clause (Db2 SQL) • order-by-clause (Db2 SQL)	No
GRANT (table or view privileges) (Db2 SQL)	New clause: UNLOAD.	No
GRANT (system privileges) (Db2 SQL)	New clause: BINDAGENT.	No
MERGE	The MERGE statement now includes the delete operation as part of the merge process. The following new clauses are added: • *signal-statement* • ELSE IGNORE The following clauses are changed: • AS *correlation-name* • *assignment-clause* • WHEN *matching-condition* • THEN *modification-operation*	No

SQL element	Change introduced	Incompatible change?
OPEN	Changed clause: USING.	No
PREPARE (Db2 SQL)	New clause: offset-clause (Db2 SQL). Changed clause: fetch-clause (Db2 SQL).	No
SELECT INTO (Db2 SQL)	New clause: offset-clause (Db2 SQL) The following clauses are changed: • INTO • fetch-clause (Db2 SQL)	No
SET assignment-statement (Db2 SQL)	Changed clause: DEFAULT.	No
subselect (Db2 SQL)	New clause: offset-clause (Db2 SQL) The following clauses are changed: • *correlation-clause* of collection-derived-table (Db2 SQL) in a table-reference of a FROM clause • fetch-clause (Db2 SQL) • A user-defined function that is defined with MODIFIES SQL DATA can be invoked.	No
TRANSFER OWNERSHIP (Db2 SQL)	New statement.	No
UPDATE (Db2 SQL)	New clause: BETWEEN *value-1* AND *value-2* clause of the *period-clause*.	No
VALUES INTO	The following clauses are changed: • Assignment clause source • INTO	No
compound-statement (Db2 SQL) for SQL routines	New clause: ATOMIC. Changed clause: DEFAULT or CONSTANT.	No

For more information about these changes, see "Function level 500 (installation or migration - October 2016)" on page 104.

SQL changes in function level 100

The following SQL changes take effect in Db2 12 for applications that run at application compatibility V12R1M100 or higher.

SQL element	Change introduced	Incompatible change?
REVOKE (table or view privileges) (Db2 SQL)	New clause: UNLOAD.	No

For more information about these changes, see "Function level 100 (migration)" on page 113.

Special register changes

GUPI Db2 12 introduces the following changes to special registers:

- FL 504 New alternative spelling for certain special registers, as shown in the following table.

Existing Special Register	New Syntax Alternative
CURRENT CLIENT_ACCTNG	CLIENT ACCTNG
CURRENT CLIENT_APPLNAME	CLIENT APPLNAME
CURRENT CLIENT_USERID	CLIENT USERID
CURRENT CLIENT_WRKSTNNAME	CLIENT WRKSTNNAME
CURRENT SERVER	CURRENT_SERVER
CURRENT TIME ZONE CURRENT TIMEZONE	CURRENT_TIMEZONE

Predicate changes

GUPI Db2 12 introduces the following changes to SQL predicates:

- FL 504 Alternative spellings for ISNULL and NOT NULL predicates, as shown in the following table.

Existing Predicate	New Syntax Alternative
IS NULL	ISNULL
IS NOT NULL	NOTNULL

- FL 500 Predicates involving *row-value-expressions* can use the following additional comparison operators: <, >, <=, and >=. See Basic predicate (Db2 SQL) for details.

GUPI

New and changed built-in functions

Db2 12 introduces or changes the following built-in functions.

Important information about existing user-defined functions: When a new application compatibility level introduces a new or changed built-in function that has the same name and signature as an existing user-defined function, unqualified references to the user-defined function might resolve incorrectly. Applications that have unqualified references to the user-defined function might fail. To avoid this situation, modify applications to explicitly qualify references to user-defined functions with the same name and signature as the new or changed built-in functions.

GUPI

APPLCOMPAT level	Function name	Change introduced	Incompatible change?
V12R1M507	Various	The following functions are newly supported in Db2 for z/OS as passthrough-only expressions, which are passed through to IBM Db2 Analytics Accelerator for z/OS. - ADD_DAYS (Db2 SQL) - BTRIM (Db2 SQL) - DAYS_BETWEEN (Db2 SQL)	No

APPLCOMPAT level	Function name	Change introduced	Incompatible change?
		• NEXT_MONTH (Db2 SQL) • REGR_AVGX • REGR_AVGY • REGR_COUNT • REGR_INTERCEPT or REGR_ICPT • REGR_R2 • REGR_SLOPE • REGR_SXX • REGR_SXY • REGR_SYY • ROUND_TIMESTAMP (Db2 SQL) if invoked with a date expression	
V12R1M506	HASH (Db2 SQL)	New built-in function.	No
V12R1M506	CHARACTER_LENGTH or CHAR_LENGTH (Db2 SQL)	CHAR_LENGTH is now supported as an alternative function name.	No
V12R1M506	CLOB (Db2 SQL)	TO_CLOB is now supported as an alternative function name.	No
V12R1M506	COVAR_POP or COVARIANCE or COVAR (Db2 SQL)	COVAR_POP is now supported as an alternative function name.	No
V12R1M506	LEFT (Db2 SQL)	STRLEFT is now supported as an alternative function name.	No
V12R1M506	POWER or POW (Db2 SQL)	POW is now supported as an alternative function name.	No
V12R1M506	POSSTR (Db2 SQL)	STRPOS is now supported as an alternative function name.	No
V12R1M506	RANDOM or RAND (Db2 SQL)	RANDOM is now supported as an alternative function name.	No
V12R1M506	RIGHT (Db2 SQL)	STRRIGHT is now supported as an alternative function name.	No
V12R1M506	TIMESTAMP_FORMAT (Db2 SQL)	TO_TIMESTAMP is now supported as an alternative function name.	No
V12R1M505	DECRYPT_DATAKEY_INTEGER, DECRYPT_DATAKEY_BIGINT, DECRYPT_DATAKEY_DECIMAL, DECRYPT_DATAKEY_VARCHAR, DECRYPT_DATAKEY_CLOB, DECRYPT_DATAKEY_VARGRAPHIC, DECRYPT_DATAKEY_DBCLO	New built-in functions.	No

APPLCOMPAT level	Function name	Change introduced	Incompatible change?
	B, and DECRYPT_DATAKEY_BIT		
V12R1M505	ENCRYPT_DATAKEY	New built-in function.	No
V12R1M504	Various	The following functions are newly supported in Db2 for z/OS as passthrough-only expressions, which are passed through to IBM Db2 Analytics Accelerator for z/OS. • CUME_DIST • CUME_DIST (aggregate) (Db2 SQL)FIRST_VALUE • LAG • LAST_VALUE • LEAD • NTH_VALUE • NTILE • PERCENT_RANK • PERCENT_RANK (aggregate) (Db2 SQL) • RATIO_TO_REPORT • REGEXP_COUNT (Db2 SQL) • REGEXP_INSTR (Db2 SQL) • REGEXP_LIKE (Db2 SQL) • REGEXP_REPLACE (Db2 SQL) • REGEXP_SUBSTR (Db2 SQL)	No
V12R1M502	GRAPHIC (Db2 SQL)	The first argument now accepts numeric data types, including SMALLINT, INTEGER, BIGINT, DECIMAL, REAL, DOUBLE, FLOAT, and DECFLOAT.	No
V12R1M502	VARGRAPHIC (Db2 SQL)	The first argument accepts numeric data types, including SMALLINT, INTEGER, BIGINT, DECIMAL, REAL, DOUBLE, FLOAT, and DECFLOAT.	No
V12R1M501	LISTAGG (Db2 SQL)	New built-in function.	No
V12R1M500	ARRAY_AGG (Db2 SQL)	Newly supported built-in function when used for associative array aggregation.	No
V12R1M500	GENERATE_UNIQUE_BINARY	New built-in function.	No
V12R1M500	HASH_CRC32, HASH_MD5, HASH_SHA1, and HASH_SHA256	New built-in functions.	No
V12R1M500	LOWER (Db2 SQL)	The following locales can now be specified: • UNI_60 • UNI_90	

APPLCOMPAT level	Function name	Change introduced	Incompatible change?
V12R1M500	PERCENTILE_CONT	New built-in function.	No
V12R1M500	PERCENTILE_DISC	New built-in function.	No
V12R1M500	TRANSLATE (Db2 SQL)	The following locales can now be specified: • UNI_60 • UNI_90	No
V12R1M500	UPPER (Db2 SQL)	The following locales can now be specified: • UNI_60 • UNI_90	No
V12R1M500	WRAP (Db2 SQL)	New built-in function.	No
V12R1M100	BLOCKING_THREADS (Db2 SQL)	New built-in function.	

GUPI

New and changed Db2-supplied stored procedures

GUPI Db2 12 introduces or changes the following Db2-supplied stored procedures:

- FL 500 CREATE_WRAPPED stored procedure (Db2 SQL) is new
- FL 500 DSNUTILV stored procedure (Db2 SQL) is new
- FL 500 DSNACCOX stored procedure (Db2 SQL): Several of the default values have changed

GUPI

Built-in global variables

GUPI Db2 12 introduces the following built-in global variables:

- FL 507 MAX_LOCKS_PER_TABLESPACE (Db2 SQL)
- FL 507 MAX_LOCKS_PER_USER (Db2 SQL)
- FL 500 TEMPORAL_LOGICAL_TRANSACTION_TIME (Db2 SQL)
- FL 500 TEMPORAL_LOGICAL_TRANSACTIONS (Db2 SQL)

GUPI

New reserved words in Db2 12

GUPI Db2 12 introduces the following SQL reserved words:

- FL 504 CURRENT_SERVER
- FL 504 CURRENT_TIMEZONE
- FL 500 LIMIT
- FL 500 OFFSET

For the complete list, see Reserved words in Db2 for z/OS (Db2 SQL).

GUPI

Related reference

Statements (Db2 SQL)

Reserved words in Db2 for z/OS (Db2 SQL)

Application compatibility levels in Db2 12

The *application compatibility level* of your applications controls the adoption and use of new capabilities and enhancements, and the impact of incompatible changes. The advantage is that you can complete the Db2 12 migration process without the need to update your applications immediately.

After function level 500 or higher is activated, you can continue run applications with the features and behavior of previous versions or specific Db2 12 function levels.

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 most SQL statements, including data definition statements (such as CREATE and ALTER statements) and data control statements (such as GRANT and REVOKE statements).

The application compatibility of a package is initially set when you bind a package, based on the following values:

1. The APPLCOMPAT bind option value, if specified.
2. If the bind option is omitted, the APPLCOMPAT subsystem parameter.

For static SQL statements, the APPLCOMPAT column of the SYSIBM.SYSPACKAGE catalog table stores the application compatibility setting. This setting changes for the following reasons:

- You issue a REBIND command for the package and specify a different value for the APPLCOMPAT option. If you omit this option, the previous value for the package is used. If no previous value is available (such as for packages last bound before the introduction of application compatibility) the APPLCOMPAT subsystem parameter value is used.

- An automatic bind of the package occurs. The application compatibility is set to the previous value. If no previous value is available, the APPLCOMPAT subsystem parameter value is used.

For dynamic SQL statements, the CURRENT APPLICATION COMPATIBILITY special register stores the application compatibility setting. This setting changes for the following reasons:

- The special register is initialized to the application compatibility of the package, as described above.

- During execution of the package, SET CURRENT APPLICATION COMPATIBILITY statements can change the special register. The value must be equivalent to the APPLCOMPAT bind option for the package or lower, if the value is V12R1M500 or above.

For new installations, the default APPLCOMPAT subsystem parameter value is V12R1M500. However, you can specify a higher value. For migrated environments, the default value is the value from the migration input member.

Tip: When you migrate to Db2 12, or activate any higher function level, change the APPLCOMPAT subsystem parameter value only after all applications can use the SQL capabilities of Db2 12 or the higher function level. For details, see Enabling default application compatibility with function level 500 or higher (Db2 Application programming and SQL).

Supported application compatibility levels

Db2 12 supports the following application compatibility levels in most contexts:

V*vv*R*r*M*mmm*

Compatibility with the behavior of the identified Db2 function level. For example, V12R1M510 specifies compatibility with the highest available Db2 12 function level. The equivalent function level or higher must be activated. For a list of supported Db2 12 function levels, see Chapter 2, "Db2 12 function levels," on page 69.

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. After migration to Db2 12, this value has the same result as specifying V12R1M100.

V10R1
Compatibility with the behavior of DB2 10 new-function mode.

Example: V10R1 application compatibility

The following example shows the results of using a capability that is introduced in the application compatibility level V11R1, with application compatibility level set to V10R1. Assume that the APPLCOMPAT subsystem parameter value is V10R1. The example CREATE PROCEDURE statement does not specify the APPLCOMPAT keyword. In this example the CREATE TYPE statement is successful but the CREATE PROCEDURE statement results in SQLCODE -4743.

```
CREATE TYPE PHONENUMBERS AS VARCHAR(12) ARRAY ??(1000000??)
```

```
DSNT400I SQLCODE = 000,  SUCCESSFUL EXECUTION
```

```
CREATE PROCEDURE FIND_CUSTOMERS(
  IN NUMBERS_IN KRAMSC01.PHONENUMBERS,
  IN AREA_CODE CHAR(3),
  OUT NUMBERS_OUT KRAMSC01.PHONENUMBERS)
    BEGIN
      SET NUMBERS_OUT =
        (SELECT ARRAY_AGG(T.NUM)
        FROM UNNEST(NUMBERS_IN) AS T(NUM)
        WHERE SUBSTR(T.NUM, 1, 3) = AREA_CODE);
    END
```

```
DSNT408I SQLCODE = -4743, ERROR:  ATTEMPT TO USE A FUNCTION WHEN THE
   APPLICATION COMPATIBILITY SETTING IS SET FOR A PREVIOUS LEVEL
```

The APPLCOMPAT bind option value for the CREATE PROCEDURE statement is then set to V11R1 or higher and result of the statement is then successful.

```
CREATE PROCEDURE FIND_CUSTOMERS(
  IN NUMBERS_IN KRAMSC01.PHONENUMBERS,
  IN AREA_CODE CHAR(3),
  OUT NUMBERS_OUT KRAMSC01.PHONENUMBERS)
    APPLCOMPAT V11R1
    BEGIN
      SET NUMBERS_OUT =
        (SELECT ARRAY_AGG(T.NUM)
        FROM UNNEST(NUMBERS_IN) AS T(NUM)
        WHERE SUBSTR(T.NUM, 1, 3) = AREA_CODE);
    END
```

```
DSNT400I SQLCODE = 000,  SUCCESSFUL EXECUTION
```

Related concepts
Function levels and related levels in Db2 12

Enhancements to Db2 12 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

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 12 migration and function level activation processes from application updates to adopt new features and resolve incompatible changes.

Related reference

APPLCOMPAT bind option (Db2 Commands)

CURRENT APPLICATION COMPATIBILITY special register (Db2 SQL)

APPL COMPAT LEVEL field (APPLCOMPAT subsystem parameter) (Db2 Installation and Migration)

SYSPACKAGE catalog table (Db2 SQL)

SET CURRENT APPLICATION COMPATIBILITY (Db2 SQL)

-ACTIVATE (Db2) (Db2 Commands)

V12R1M*nnn* application compatibility levels

In Db2 12, you can use the application compatibility level to control the adoption of new SQL capabilities and enhancements of particular function 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 are specified in commands and message output by nine-character strings that correspond to the Db2 version, release, and modification value of the corresponding function level. See the activation details for each function level for a summary the new features that are controlled by the corresponding application compatibility level.

For example, V12R1M510 specifies compatibility with the highest available Db2 12 function level. The equivalent function level or higher must be activated. For a list of supported Db2 12 function levels, see Chapter 2, "Db2 12 function levels," on page 69.

Related reference

APPLCOMPAT bind option (Db2 Commands)

APPL COMPAT LEVEL field (APPLCOMPAT subsystem parameter) (Db2 Installation and Migration)

Setting application compatibility levels for data server clients and drivers

IBM data server clients and drivers that use Db2 for z/OS capabilities with a function level requirement of greater than V12R1M500 require extra program preparation steps.

Before you begin

Take these actions on the clients that connect to your Db2 for z/OS system:

- Determine whether you need to upgrade the data server clients or drivers to support V12R1M501 application compatibility:
  - If your applications include function that requires a minimum application compatibility level of V12R1M501, you need to upgrade to Db2 Connect Version 11.1 Modification 2 Fix Pack 2 or later.
  - If your applications include function that requires a minimum application compatibility level of V12R1M500 or earlier, you can use any Db2 Connect version.

  The minimum data server client or driver levels for exploitation of application compatibility of V12R1M501 or later are:
  - IBM Data Server Driver for JDBC and SQLJ: Versions 3.72 and 4.23, or later. For information on the driver versions that are delivered with each Db2 Connect version, see IBM Data Server Driver for JDBC and SQLJ versions and Db2 or Db2 Connect levels (Db2 Application Programming for Java).
  - Other IBM data server clients and drivers: Db2 for Linux, UNIX, and Windows, Version 11.1 Modification 2 Fix Pack 2, or later.
- Run the db2connectactivate utility to activate the Version 11.1 license certificate file for Db2 Connect Unlimited Edition on Db2 for z/OS. Specify the options to bind the driver and client packages into the NULLID collection, with APPLCOMPAT V12R1M500. For example:

```
db2connectactivate.sh -host sys1.svl.ibm.com -port 5021 -database STLEC1 -user dbadm
-password dbadmpass -bindoptions "APPLCOMPAT V12R1M500" -collection NULLID
```

For more information, see:

> Activating the license key for Db2 Connect Unlimited Edition (IBM Z)
> db2connectactivate - Server license activation utility

About this task

This procedure sets the V12R1M*nnn* application compatibility for a client or driver that needs to utilize server capabilities that require an application compatibility of greater than V12R1M500. If the client or driver does not utilize the capabilities of a new function level, you do not need to update its application compatibility settings.

Procedure

1. Set the Db2 for z/OS server application compatibility (APPLCOMPAT) level.
   a) For client applications that contain static SQL statements, rebind the static application packages with the new APPLCOMPAT value, on the client and on the server.
   b) For client applications that contain only dynamic SQL statements, bind or rebind the client or driver packages with the new APPLCOMPAT value, on the server.

   Tip: Binding package copies and keeping the original driver packages lets you access new capabilities for applications that need them, while ensuring stability for applications that should not be exposed to incompatibilities.

   For drivers only, you can use jobs that are provided with Db2 for z/OS, in data set *prefix*.SDSNSAMP, to bind or rebind the driver packages on the server. To run those jobs, follow these steps:

i) Customize jobs <u>DSNTIJLC</u> and <u>DSNTIJLR</u>, using the instructions in the job prologs.

ii) If there is a possibility that you still need to run applications under a driver that is at the old application compatibility level, run DSNTIJLC to bind copies of the driver packages at the new application compatibility level, while leaving the packages in the NULLID collection at the old application compatibility level. In most cases, you should do this.

If you are sure that you do not need to run applications under a driver at the old application compatibility level, run job DSNTIJLR to rebind the client or driver packages at the new application compatibility level.

iii) If you bound copies of the driver packages for the new function level, switch the drivers to the new function level by modifying the property that controls the current package set to match the collection ID of the new package copies.

- For CLI or ODBC drivers, change the CLI/ODBC CurrentPackageSet configuration keyword value.
- For the IBM Data Server Driver for JDBC and SQLJ, change the `DB2BaseDataSource.currentPackageSet Connection` or `DataSource` property value.

2. Set the client application compatibility value to control the capabilities of client applications when a client or driver contains changes that enable new server capabilities. If you set the client application compatibility level, its value must be less than or equal to the server application compatibility level.

Take one of the following actions to set the client application compatibility value.

- For CLI or ODBC drivers or IBM Data Server clients, change the CLI/ODBC ClientApplCompat configuration keyword value.

Do this by adding a line similar to this example to the <databases> section or the <dsn> section in the db2dsdriver.cfg file.

```
<parameter name="clientApplCompat" value="V12R1M501"/>
```

For more information, see:

- <u>IBM data server driver configuration file</u>
- <u>Installing the IBM Data Server Driver Package software on the Linux and UNIX operating systems</u> (includes information on creating and populating the db2dsdriver.cfg file)
- <u>ClientApplCompat IBM data server driver configuration keyword</u>
- For the IBM Data Server Driver for JDBC and SQLJ, change the `DB2BaseDataSource.clientApplcompat Connection` or `DataSource` property value.

Related tasks
<u>Using profile tables to control which Db2 for z/OS application compatibility levels to use for specific data server client applications (Db2 Application programming and SQL)</u>
Related reference
<u>IBM Data Server Driver for JDBC and SQLJ properties for Db2 for z/OS (Db2 Application Programming for Java)</u>
<u>-DISPLAY LOCATION (Db2) (Db2 Commands)</u>
<u>-ACTIVATE (Db2) (Db2 Commands)</u>
Related information
<u>-30025 (Db2 Codes)</u>
<u>DSNL200I (Db2 Messages)</u>

V11R1 application compatibility level

When you set the application compatibility level to V11R1, applications that attempt to use functions and features that are introduced in Db2 12 or later might behave differently or receive an error.

When new function is activated in your Db2 12 environment, you can run individual applications with some of the features and behavior of Db2 11. That is, your applications can continue to experience V11R1 behavior after new function is activated in Db2 12 or later. Then, you can migrate each application to a new application compatibility value separately until all are migrated. If application compatibility level is set to V11R1 and you attempt to use the new functions of a later version, SQL might behave differently or result in a negative SQLCODE, such as SQLCODE -4743.

PSPI

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

PSPI

A migrated Db2 12 environment behaves with V11R1 application compatibility until function level 500 or higher is activated.

The following table shows many of the features and functions that are controlled by application compatibility, and the results if you specify V11R1. If a behavior difference is traced, then the IFCID trace function code is shown.

Table 7. Behavior of V11R1 application compatibility

Feature or Function	Result with V11R1 application compatibility	IFCID 0376 trace function code
The POWER built-in function returns a result with the DOUBLE data type. The result is out of range.	SQLCODE -802	1201
CURRENT_SERVER or CURRENT_TIMEZONE is used as a column name or variable name.	SQLCODE -206	1204

Related information
SQL Reference (Db2 11 for z/OS)
Application Programming and SQL Guide (Db2 11 for z/OS)

V10R1 application compatibility level

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

In Db2 12, you can continue to run individual applications with some of the features and behavior of DB2 10. That is, your applications can continue to experience V10R1 behavior while in Db2 12, regardless of whether new function is activated. Then, you can migrate each application to a new application compatibility value separately until all are migrated. If application compatibility is set to V10R1 and you attempt to use the new functions of a later version, SQL might behave differently or result in a negative SQLCODE, such as SQLCODE -4743.

PSPI

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

◁ PSPI

A migrated Db2 12 environment behaves with V11R1 application compatibility until function level 500 or higher is activated. Application and SQL incompatibilities are described in the migration information for each version.

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

Table 8. Behavior of V10R1 application compatibility

Feature or Function	Result with V10R1 application compatibility	IFCID 0376 trace function code
An SQL statement in a client application includes an unsupported conversion (from a string type to a numeric type or from a numeric type to a string type), and implicit casting is disabled (DDF_COMPATIBILITY is set to SP_PARMS_NJV or to DISABLE_IMPCAST_NJV).	SQLCODE -301	7[1] on page 170
A client application executes an SQL CALL statement to execute a Db2 for z/OS stored procedure. The DDF_COMPATIBILITY subsystem parameter is set to SP_PARMS_NJV for client applications other than Java applications, or SP_PARMS_JV for Java applications.	The data types of the data that is returned from the SQL CALL statement match the data types of the CALL statement arguments. This behavior is compatible with the behavior before Version 10.	8[1] on page 170
A client application accesses Db2 11 from an IBM Data Server Driver for JDBC and SQLJ client. The DDF_COMPATIBILITY subsystem parameter is set to IGNORE_TZ for Java applications.	The Db2 server ignores the TIMEZONE portion, appended by the IBM Data Server Driver for JDBC and SQLJ, of the value in the TIMESTAMP WITH TIMEZONE input to a TIMESTAMP target. This behavior is compatible with the behavior before DB2 10.	9
BIF_COMPATIBILITY is set to V9_TRIM, and input *string-expression* is EBCDIC mixed data for the RTRIM, LTRIM, or STRIP built-in function.	The DB2 9 version of SYSIBM.LTRIM(*string-expression*), SYSIBM.RTRIM(*string-expression*), or SYSIBM.STRIP(*string-expression*) is executed.	10
An implicit insert or update of an XML document node	SQLCODE -20345	1101
A predicate expression with an explicit cast or an operation with an invalid value that does not affect the results of XPath processing	SQLCODE -20345	1102
How the resource limit facility uses ASUTIME value for nested routines	SQLCODE -905 is issued only when the ASUTIME limit of the top-level calling package is encountered.	1103

Table 8. Behavior of V10R1 application compatibility (continued)

Feature or Function	Result with V10R1 application compatibility	IFCID 0376 trace function code
The lengths of values that are returned from CURRENT CLIENT_USERID, CURRENT CLIENT_WRKSTNNAME, CURRENT CLIENT_APPLNAME, or CURRENT CLIENT_ACCTNG special register are longer than the DB2 10 limits.	The special register values are truncated to the DB2 10 maximum lengths and padded with blanks	1104, 1105, 1106, 1107
A CAST(string as TIMESTAMP) specification with an input string of length of 8 or an input string of length 13	An explicit cast specification from string as TIMESTAMP interprets an 8-byte character string as a Store Clock value and a 13-byte string as a GENERATE_UNIQUE value. CAST result might be incorrect.	1109
Invocation of the SPACE or VARCHAR built-in function when the result is defined as VARCHAR(32765), VARCHAR(32766), or VARCHAR(32767)	No error	1110, 1111
Subsystem parameter XML_RESTRICT_EMPTY_TAG is set to YES, and an empty XML element is serialized as *<emptyElement></emptyElement>*	No error	1112
Specification of bind option DBPROTOCOL(DRDACBF)	DSNT298I	
A period specification that follows the name of a view in the FROM clause of a query	SQLCODE -4743	
A period clause that follows the name of a target view in an UPDATE or DELETE statement	SQLCODE -4743	
A SET CURRENT TEMPORAL SYSTEM_TIME statement	SQLCODE -4743	
A SET CURRENT TEMPORAL BUSINESS_TIME statement	SQLCODE -4743	
A reference to a global variable	SQLCODE -4743	
Use of array operations and built-in functions such as • Use of the UNNEST collection-derived-table • Use of the ARRAY_FIRST, ARRAY_LAST, ARRAY_NEXT, ARRAY_PRIOR, ARRAY_AGG, TRIM_ARRAY, CARDINALITY, MAX_CARDINALITY built-in functions • A SET assignment-statement of an array element as a target table • A CAST specification with a parameter marker as the source and an array as the data type	SQLCODE -4743	
An aggregate function that contains the keyword DISTINCT and references a column that is defined with a column mask	SQLCODE -20478	

Table 8. Behavior of V10R1 application compatibility (continued)

Feature or Function	Result with V10R1 application compatibility	IFCID 0376 trace function code
An SQL statement contains the GROUP BY clause and references a column that is defined with a column mask	SQLCODE -20478	
An SQL statement contains the set operator UNION ALL or UNION DISTINCT and references a column that is defined with a column mask	SQLCODE -20478	
A reference to an alias for a sequence object	SQLCODE -4743	
A reference to an unqualified sequence that is not resolved to a public alias	SQLCODE -204	
A SELECT with a table function reference that includes a typed correlation clause	SQLCODE -4743	
A *table-reference*, *collection-derived-table*, or *xmltable-expression* that does not include a correlation-clause.	SQLCODE -4743	
A CALL statement that specifies an autonomous procedure	SQLCODE -4743	
The following datetime assignments: • A valid string representation of a timestamp to a date column • A valid string representation of a timestamp to a time column • A valid string representation of a date to a timestamp column	SQLCODE -180	

Notes:

1. PSPI To find details about the incompatible parameters, examine the contents of fields QW0376SC_Var, QW0376PR_Var, and QW0376INC_Var. See the DSNWMSGS file for more information. PSPI

Related concepts
V11R1 application compatibility level (Db2 Application programming and SQL)
Related information
SQL Reference (DB2 10 for z/OS)
Application programming and SQL Guide (DB2 10 for z/OS)

Managing application incompatibilities

Before you move an application to a new application compatibility level, you need to find application incompatibilities, adjust your applications for those incompatibilities, and verify that the incompatibilities no longer exist.

Procedure

1. Start a trace that includes IFCID 0239 to capture package information.

For example, issue the following START TRACE command: `GUPI`

```
-START TRACE(ACCTG) CLASS(7,8,10)
```

`GUPI`

2. Examine the trace output.

   `PSPI` IFCID 0239 field QPACFLGS contains a bit that is on if a package contains incompatibilities. If this bit is off, no incompatibilities were detected, and you can skip the rest of the steps. If this bit is on, proceed to step 3. `PSPI`

3. Start a trace for IFCID 0376 to report incompatibility information about the packages.

   For example, issue the following START TRACE command: `GUPI`

```
-START TRACE(PERFM) CLASS(32) IFCID(376)
```

`GUPI`

4. Run the application.

5. Examine the trace output.

   `PSPI` IFCID 0376 fields contain information about the incompatibilities. Db2 writes a single trace record for each SQL statement that is incompatible with the subsequent Db2 function level. See file *prefix*.SDSNIVPD(DSNWMSGS) for listings of the IFCID 0239 and 0376 trace records. `PSPI`

6. Revise the application to avoid any application incompatibilities.

7. Prepare the application for execution. When you bind the packages for the application, use the old APPLCOMPAT value.

8. Run the application.

9. Examine the trace output again to verify that the incompatibilities no longer exist.

What to do next

When the application runs at the old level with no application incompatibilities, rebind the package with the APPLCOMPAT value for the new function level.

Related concepts

Performance trace (Db2 Performance)

Related reference

-START TRACE (Db2) (Db2 Commands)

Trace field descriptions (Db2 Performance)

Enabling default application compatibility with function level 500 or higher

The APPLCOMPAT subsystem parameter specifies the default value of the APPLCOMPAT bind option. Before function level 500 or higher is activated, the APPLCOMPAT subsystem parameter must be set to V11R1 or V10R1. These settings ensure that existing SQL applications are bound for compatibility with the earlier release by default.

Before you begin

1. Activate function level 500 or higher, as described in Activating Db2 12 new function at migration (Db2 Installation and Migration).

2. For any packages that need to continue running at a lower level, bind or rebind them and explicitly specify the APPLCOMPAT bind option. For more information, see Chapter 15, "Controlling the Db2 application compatibility level," on page 289.

3. Take the following precautions to ensure that applications are ready to run at the higher application compatibility level by default.

- Identify and resolve all application incompatibilities of the higher level, as described in Managing application incompatibilities (Db2 Application programming and SQL).
- Rebind any packages that must continue to run at the lower application compatibility level and explicitly specify the APPLCOMPAT bind option for that level.

About this task

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).

Procedure

To enable default application compatibility with the current function level:

1. Change the APPLCOMPAT subsystem parameter setting. Set the value to V12R1M500 or the equivalent higher active *function-level* value.

   You can complete this step as described in Updating subsystem parameter and application default values (Db2 Installation and Migration), or by modifying your customized copy of the DSNTIJUZ job.

   The format is V*vv*R*r*M*mmm*, where *vv* is the version, *r* is the release, and *mmm* is the modification level. For example, V12R1M510 identifies function level 510. For a list of all available function levels in Db2 12, see Chapter 2, "Db2 12 function levels," on page 69. See the activation details for each function level for a summary the new features that are controlled by the corresponding application compatibility level.

2. Run the first two job steps of DSNTIJUZ to rebuild your subsystem parameter (DSNZP*xxx*) module.
3. Use the -SET SYSPARM command or restart Db2.

Results
Future bind and rebind operations set the application compatibility level of the package to the APPLCOMPAT subsystem parameter value, if the APPLCOMPAT bind option is not specified. Packages that are bound or rebound at the higher level can begin use of SQL capabilities introduced at that level.
Related concepts
Application compatibility levels in Db2 12 (Db2 Application programming and SQL)
Related tasks
Adopting new capabilities in Db2 12 continuous delivery
In Db2 12, function levels and application compatibility levels control the adoption of most new capabilities by Db2 subsystems and Db2 applications.

Activating Db2 12 new function at migration (Db2 Installation and Migration)
Related reference
APPL COMPAT LEVEL field (APPLCOMPAT subsystem parameter) (Db2 Installation and Migration)
APPLCOMPAT bind option (Db2 Commands)

Utility changes in Db2 12

You can use this information to plan for utility changes in Db2 12.

Important: Db2 12 introduces continuous delivery of new capabilities and enhancements in function levels. Most new capabilities become available only after activation of the Db2 12 function level that introduces them, or when applications run with the corresponding application compatibility level. For more information, see Chapter 12, "Activating Db2 12 function levels," on page 281.

Utility changes in function level 501 and higher

The following changes take effect when you activate function level 501 or higher in Db2 12. The changes are listed in function level order with the highest at the top.

Function level	Utility	Change introduced	Incompatible change?
FL 507	LOAD	If you specify the USE PROFILE option with inline statistics, Db2 deletes statistics that are not specified in the profile.	No
FL 507	RUNSTATS	If you specify the USE PROFILE option, Db2 deletes statistics that are not specified in the profile.	No

Utility changes in function level 500

The following changes take effect when you activate function level 500 in Db2 12.

Utility	Change introduced	Incompatible change?
MODIFY RECOVERY	The following new options are added: • DELETEDS • FLASHCOPY ONLY • NOCOPYPEND When MODIFY RECOVERY is run at the table space level, a SYSIBM.SYSCOPY row with ICTYPE='M' and STYPE='R' is inserted for each partition of the table space and any partitioned indexes with the COPY YES attribute that the MODIFY utility processes. This differs from previous versions where only one SYSCOPY record was inserted for the entire table space, regardless of partitions.	No
RECOVER	FROM	No
REORG TABLESPACE	The following new options are added: • DROP_PART NO • DROP_PART YES • ICLIMIT_DASD • ICLIMIT_TAPE	No
TEMPLATE	The BLKSZLIM option is added.	No
UNLOAD (Db2 Utilities)	The authorization checking is changed, so that Db2 no longer checks for the SELECT privilege on the tables being unloaded.	Yes

Utility changes in function level 100

The following changes take effect when you migrate to function level 100 in Db2 12.

Utility	Change introduced	Incompatible change?
BACKUP SYSTEM	New options: • ALTERNATE_CP • DBBSG • LGBSG	No
CATMAINT	New option: LEVEL. (Specify a function level or catalog level. If you specify a function level Db2 determines the appropriate catalog level.)	No
CATENFM	With the introduction of single-phase migration in Db2 12, the CATENFM utility is not needed for the migration process and is removed.	No
DSN1COMP	New options: • COMPTYPE • LOB	No
DSNJU008 (print CDDS) (Db2 Utilities)	New utility.	No
DSN1LOGP	New option: PART.	No
DSN1PRNT	New option: PART.	No
LOAD	The following new options are added: • CCSID in a field specification • DEFINEAUX • DRAIN_WAIT • FORCE • IGNORE • INVALIDATECACHE • KEEP_EMPTY_PAGES • NOCHECKPEND • OVERRIDE(ROWCHANGE) • RETRY • PRESORT • RETRY_DELAY • SWITCHTIME • STATCLGMEMSRT • STATCLGMEMSRT • UPDMAXASSIGNEDVAL The following options are changed: • You can now request that the LOAD utility take partition-level sequential image copies by specifying the copy options only once, at the table space level.	Yes

Utility	Change introduced	Incompatible change?
	• If the LOAD_RO_OBJECTS subsystem parameter setting is YES, you can load data into read-only objects. • You can specify multiple SYSREC data sets by providing multiple *ddname* values in a list. The data sets represented by those *ddname* values are dynamically concatenated to a single *ddname*, which becomes the input to LOAD. • A restriction against use of the COLGROUP keyword in conjunction with PARALLEL keyword is removed. In Db2 11, you can collect cardinality, frequency, and histogram statics for column groups when the PARALLEL keyword is specified. • DEFAULTIF is enhanced to accept <> (not equal) and CONV_ERROR (conversion error). • FREQVAL with MOST, LEAST, or BOTH is supported for indexes. • Support is added for new IGNORE options: PART, CONV, VALPROC, IDERROR, DUPKEY • Support is added for concurrent access to target tables during LOAD utility processing, with REPLACE SHRLEVEL REFERENCE. • Support for a wider range of external date formats is added with DATE EXTERNAL (*date-format*) • Support for a wider range of external time formats is added with TIME EXTERNAL (*time-format*) • For inline statistics, if you specify the FREQVAL keyword and omit the COUNT keyword, Db2 automatically determines the number of frequently occurring values to collect so that the data distribution is no longer skewed. The LOAD utility now always maintains the value of MAXASSIGNEDVAL in SYSIBM.SYSEQUENCES for any loaded identity columns. The RRF subsystem parameter is removed, and the default behavior is RRF=ENABLE, which means that: • Newly created table spaces or newly added partitions on partition-by-growth table spaces are created in the reordered row format. • If LOAD REPLACE does not specify the ROWFORMAT BRF option, those utilities convert existing basic row format table spaces to reordered row format.	
REBUILD INDEX	The following new options are added:	No

Utility	Change introduced	Incompatible change?
	- BOTH - LEAST - MOST - INVALIDATECACHE For inline statistics, if you specify the FREQVAL keyword and omit the COUNT keyword, Db2 automatically determines the number of frequently occurring values to collect so that the data distribution is no longer skewed.	
RECOVER	The following new options are added: - ALTERNATE_CP - FLASHCOPY_PPRCP - INLCOPY - NOSYSCOPY - KEEP - SCOPE	Yes
REORG INDEX	The following new options are added: - BOTH - LEAST - LASTLOG - MOST - NOSYSUT1 - PARALLEL - INVALIDATECACHE For inline statistics, if you specify the FREQVAL keyword and omit the COUNT keyword, Db2 automatically determines the number of frequently occurring values to collect so that the data distribution is no longer skewed.	No
REORG TABLESPACE	The following new options are added: - INITCDDS - INVALIDATECACHE - LASTLOG - NOCHECKPEND - SEARCHTIME - STATCLGMEMSRT For inline statistics, if you specify the FREQVAL keyword and omit the COUNT keyword, Db2 automatically determines the number of frequently occurring values to collect so that the data distribution is no longer skewed.	Yes

Utility	Change introduced	Incompatible change?
	The REORG utility processes now partition-by-growth table spaces differently than in prior versions: • REORG no longer sets COPY-pending status on newly created LOB table spaces during the LOG phase. Instead, REORG will create inline image copy data sets for those LOB table spaces. • REORG will no longer fail during a reorganization of a partition-by-growth table space when it is unable to extend the partition range. Instead, REORG will now create a new partition for the partition-by-growth table space as long as the maximum number of partitions have not been reached. • REORG TABLESPACE has a new keyword, DROP_PART NO or DROP_PART YES, that can be specified to delete the highest numbered partitions if they are empty when REORG successfully completes. In prior versions, this behavior could only be accomplished using the REORG_DROP_PBG_PARTS subsystem parameter was set to ENABLE. In addition to the REORG changes for processing partition-by-growth table spaces, REORG in Db2 12 will fail if it is using a FlashCopy and the creation of the online FlashCopy image fails. Also, for REORG TABLESPACE, when SHRLEVEL REFERENCE or SHRLEVEL CHANGE is specified and blocking claimers are encountered, the DISPLAY DATABASE command output will now identify the blocking claimers on each and every drain attempt failure. In prior versions, only the most recent drain attempt failure was identified. Also, the format of mapping tables has changed to accommodate the 7-byte RIDS that are needed for partition-by-range table spaces with relative page numbering. Also, the RRF subsystem parameter is removed, and the new default behavior is RRF=ENABLE, which means that: • Newly created table spaces or newly added partitions on partition-by-growth table spaces are created in the reordered row format. • If LOAD REPLACE does not specify the ROWFORMAT BRF option, those utilities convert existing basic row format table spaces to reordered row format.	

Utility	Change introduced	Incompatible change?
	SORTNPSI YES is changed so that when you specify this option, non-partitioned secondary index (NPSI) keys are sorted.	
REPAIR	The following new options are added: • WRITELOG • INSERTVERSIONPAGES • SETCURRENTVERSION Changed option: The VERSIONS option is no longer supported, and is replaced by the CATALOG option.	No
RESTORE SYSTEM	The following new options are added: • ALTERNATE_CP • FLASHCOPY_PPRCP • RESTOREBEFORE RESTORE SYSTEM now fails if either of the following situations occur: • The most recent system-level backup before the recovery point is not registered in the DFSMShsm repository. • An error is returned by DFSMShsm during the restore of the production volumes.	Yes
RUNSTATS	The following new options are added: • INVALIDATECACHE • REGISTER • STATCLGMEMSRT RUNSTATS now collects frequency and histogram statistics for XML indexes in the following catalog table columns: • KEYVALUE, FREQUENCYF, and TYPE in SYIBM.SYSINDEXES • CARDF, FREQUENCYF, HIGHVALUE, LOWVALUE, TYPE, QUANTILENO in SYSIBM.SYSKEYTGTDIST Db2 can use frequency and histogram statistics for XML indexes to estimate the filter factor of XMLEXISTS predicates when the following conditions are true: • The predicate is as equality (=) or range (>, >=, >=, <=) predicates • The right side of the predicate is a literal value. • Frequency or histogram statistics are collected for the XML index that matches the XMLEXISTS predicate. For more information, see Collection of statistics on XML objects (Db2 Utilities).	No

Utility	Change introduced	Incompatible change?
UNLOAD	The following new options are added: • CCSID in a field specifications • REGISTER	Yes

For information about the new options, see the information for the utility. For information about an option with a changed default value, see Utility release incompatibilities.

Related concepts
Basic information about Db2 utilities (Db2 Utilities)

Catalog changes in Db2 12

Use this information to plan for catalog changes in Db2 12.

Important: Db2 12 introduces continuous delivery of new capabilities and enhancements in function levels. Most new capabilities become available only after activation of the Db2 12 function level that introduces them, or when applications run with the corresponding application compatibility level. For more information, see Chapter 12, "Activating Db2 12 function levels," on page 281.

Db2 12 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 12 function levels use the following Db2 catalog levels:

• V12R1M509
• V12R1M507
• V12R1M505
• V12R1M503
• V12R1M502
• V12R1M500

For more information, see Chapter 8, "Function levels and related levels in Db2 12," on page 265.

Catalog changes in catalog level 501 and higher

GUPI

The following catalog changes take effect in Db2 12 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?
V12R1M509	SYSTABLEPART catalog table (Db2 SQL)	New column: COMPRESS_USED Changed column: COMPRESS	No
V12R1M509	SYSTABLESPACE catalog table (Db2 SQL)	Changed column: COMPRESS	No
V12R1M507	MAX_LOCKS_PER_TABLE_SPACE	New built-in global variable.	No

Catalog level	Catalog object	Change introduced	Incompatible change?
V12R1M507	MAX_LOCKS_PER_USER	New built-in global variable.	No
V12R1M505	SYSPACKAGE catalog table (Db2 SQL)	Previously defined unused column now used: COPYID.	No
V12R1M505	SYSPACKDEP catalog table (Db2 SQL)	Previously defined unused column now used: COPYID.	No
V12R1M505	SYSPACKSTMT catalog table (Db2 SQL)	Previously defined unused column now used: COPYID.	No
V12R1M503	REPLICATION_OVERRIDE	New built-in global variable.	
V12R1M502	SYSINDEXES catalog table (Db2 SQL)	New column: KEYLABEL.	No
V12R1M502	SYSSTOGROUP catalog table (Db2 SQL)	New column: KEYLABEL.	No
V12R1M502	SYSTABLESPACE catalog table (Db2 SQL)	New column: KEYLABEL.	No
V12R1M502	SYSTABLES catalog table (Db2 SQL)	New column: KEYLABEL.	No

Catalog changes for function level 100

The following changes catalog take effect when you tailor the catalog Db2 for migration to Db2 12 function level 100 (catalog level V12R1M500).

Catalog object	Change introduced	Incompatible change?
CATALOG_LEVEL	New built-in global variable.	No
DEFAULT_SQLLEVEL	New built-in global variable.	No
PRODUCTID_EXT	New built-in global variable.	No
SYSDYNQRY catalog table (Db2 SQL)	New table. New indexes: • DSNDQX01 • DSNDQX02 • DSNDQX11	No
SYSDYNQRYDEP catalog table (Db2 SQL)	New table. New indexes: • DSNDQX03 • DSNDQX04 • DSNDQX05 • DSNDQX12	No

Catalog object	Change introduced	Incompatible change?
SYSDYNQRY_EXPL catalog table (Db2 SQL)	New table. New index: DSNDQX08.	No
SYSDYNQRY_OPL catalog table (Db2 SQL)	New table. New index: DSNDQX10.	No
SYSDYNQRY_SHTEL catalog table (Db2 SQL)	New table. New index: DSNDQX09.	No
SYSDYNQRY_SPAL catalog table (Db2 SQL)	New table. New index: DSNDQX07.	No
SYSDYNQRY_TXTL catalog table (Db2 SQL)	New table. New index: DSNDQX06.	No
SYSINDEXCONTROL catalog table (Db2 SQL)	New table.	No
SYSIBM.SYSIXSPACESTATS_H catalog table	New table. See Temporal versioning for Db2 catalog tables (Db2 SQL).	No
SYSLEVELUPDATES catalog table (Db2 SQL)	New table.	No
SYSSESSION catalog table (Db2 SQL)	New table.	No
SYSSESSION_EX catalog table (Db2 SQL)	New table.	No
SYSSESSION_STATUS catalog table (Db2 SQL)	New table.	No
SYSAUDITPOLICIES catalog table (Db2 SQL)	New columns: • SYS_START[1] on page 188 • SYS_END[1] on page 188 • TRANS_START[1] on page 188	No
SYSCOLUMNS catalog table (Db2 SQL)	New column: ENCODING_SCHEME. Changed column: PERIOD. New index: DSNDCX06.	No
SYSCOLAUTH catalog table (Db2 SQL)	New columns: • SYS_START[1] on page 188 • SYS_END[1] on page 188 • TRANS_START[1] on page 188	No
SYSCONTEXTAUTHIDS catalog table (Db2 SQL)	New columns: • SYS_START[1] on page 188 • SYS_END[1] on page 188 • TRANS_START[1] on page 188	No

Catalog object	Change introduced	Incompatible change?
SYSCONTEXT catalog table (Db2 SQL)	New columns: • SYS_START[1] on page 188 • SYS_END[1] on page 188 • TRANS_START[1] on page 188	No
SYSCONTROLS catalog table (Db2 SQL)	New columns: • SYS_START[1] on page 188 • SYS_END[1] on page 188 • TRANS_START[1] on page 188 • REGENERATETS	No
SYSCONTROLS_DESC_H catalog table	New table for LOB data in the DESCRIPTOR column of SYSCONTROLS_H. New index: DSNTRX04.	No
SYSCONTROLS_RTXT_H catalog table	New table for LOB data in the RULETEXT column of SYSCONTROLS_H. New index: DSNTRX03.	No
SYSCOPY catalog table (Db2 SQL)	Changed columns: • DSVOLSER • ICTYPE • STYPE • TTYPE	No
SYSCTXTTRUSTATTRS catalog table (Db2 SQL)	New columns: • SYS_START[1] on page 188 • SYS_END[1] on page 188 • TRANS_START[1] on page 188	No
SYSDBAUTH catalog table (Db2 SQL)	New columns: • SYS_START[1] on page 188 • SYS_END[1] on page 188 • TRANS_START[1] on page 188	No
SYSDEPENDENCIES catalog table (Db2 SQL)	New column: DTYPE	No
SYSENVIRONMENT catalog table (Db2 SQL)	New columns: • CREATEDTS • APPLCOMPAT	No
SYSINDEXSPACESTATS catalog table (Db2 SQL)	New columns: • GETPAGES • SYS_START • SYS_END	No

Catalog object	Change introduced	Incompatible change?
	• TRANS_START Changed column: GETPAGES.	
SYSINDEXES catalog table (Db2 SQL)	New columns: • DSSIZE • PAGENUM • PARKEYCOLNUM [1] on page 188 • STATUS [1] on page 188 • INDEXSTATUS [1] on page 188 • PARTITIONS [1] on page 188 • PQTY • STORTYPE • STORNAME • VCATNAME • FREEPAGE • PCTFREE • GBPCACHE • SECQTYI • ENFORCED_CONS • IMPLICIT • REGENERATETS	No
SYSINDEXPART catalog table (Db2 SQL)	New columns: • DSSIZE • PAGENUM • LIMITKEY_EXTERNAL [1] on page 188 Changed column: SPACE.	No
SYSKEYCOLUSE catalog table (Db2 SQL)	Changed column: PERIOD.	No
SYSKEYS catalog table (Db2 SQL)	New column: CREATEDTS. Changed column: PERIOD.	No
SYSKEYTGTDIST catalog table (Db2 SQL)	Changed column: NUMKEYS.	No
SYSLEVELUPDATES catalog table (Db2 SQL)	New table. New indexes: • DSNLVX01 • DSNLVX02	No.
SYSPENDINGDDL catalog table (Db2 SQL)	New columns: • REORG_SCOPE_LOWPART	No

Catalog object	Change introduced	Incompatible change?
	• REORG_SCOPE_HIGHPART Changed column: PARTITION.	
SYSPACKAGE catalog table (Db2 SQL)	New columns: • ORIGIN • APREUSE_NO_FL • APREUSE_NO_TS • CONC_STMT • FUNCTION_LVL Changed columns: • TYPE • HOSTLANG • VERSION • DEFERPREPARE • APPLCOMPAT • KEEPDYNAMIC	No
SYSPACKAUTH catalog table (Db2 SQL)	New columns: • SYS_START[1] on page 188 • SYS_END[1] on page 188 • TRANS_START[1] on page 188	No
SYSPACKCOPY catalog table (Db2 SQL)	• ORIGIN • APREUSE_NO_FL • APREUSE_NO_TS • CONC_STMT • FUNCTION_LVL	No
SYSPACKDEP catalog table (Db2 SQL)	Changed column: DTYPE	No
SYSPACKSTMT catalog table (Db2 SQL)	New columns: • QUERYID • QUERY_HASH • QUERY_HASH_VERSION	No
SYSPLANAUTH catalog table (Db2 SQL)	New columns: • SYS_START[1] on page 188 • SYS_END[1] on page 188 • TRANS_START[1] on page 188	No
SYSQUERY catalog table (Db2 SQL)	New index: DSNQYX04.	No
SYSRELS catalog table (Db2 SQL)	Changed column: CHECKEXISTINGDATA.	No

Catalog object	Change introduced	Incompatible change?
SYSRESAUTH catalog table (Db2 SQL)	New columns: • SYS_START[1] on page 188 • SYS_END[1] on page 188 • TRANS_START[1] on page 188	No
SYSROUTINES catalog table (Db2 SQL)	New columns: • WRAPPED • REGENERATETS	No
SYSSCHEMAAUTH catalog table (Db2 SQL)	New columns: • SYS_START[1] on page 188 • SYS_END[1] on page 188 • TRANS_START[1] on page 188	No
SYSSEQUENCEAUTH catalog table (Db2 SQL)	New columns: • SYS_START[1] on page 188 • SYS_END[1] on page 188 • TRANS_START[1] on page 188	No
SYSSEQUENCES catalog table (Db2 SQL)	Changed column: MAXASSIGNEDVAL.	Yes
SYSSESSION catalog table (Db2 SQL)	New table. New index: DSNSNX02.	No
SYSSESSION_EX catalog table (Db2 SQL)	New table. New indexes: • DSNSNX04 • DSNSNX05	No
SYSSESSION_GV catalog table	New table for LOB data in the GLOBAL_VARIABLES column of the SYSSESSION table. New index: DSNSNX01.	No
SYSSESSION_DATA catalog table	New table for LOB data in the DATA column of the SYSSESSION_EX table. New index:	No
SYSSESSION_STATUS catalog table (Db2 SQL)	New index: DSNSNX06.	No
SYSTABAUTH catalog table (Db2 SQL)	New columns: • UNLOADAUTH • SYS_START [1] on page 188 • SYS_END [1] on page 188 • TRANS_START [1] on page 188	No

Catalog object	Change introduced	Incompatible change?
SYSTABCONST catalog table (Db2 SQL)	Changed column: TYPE.	No
SYSTABLEPART catalog table (Db2 SQL)	New columns: • TYPE • PAGENUM • BPOOL • PGSIZE • DSSIZE • MEMBER_CLUSTER • COMPRESSRATIO Changed column: SPACE.	No
SYSTABLESPACE catalog table (Db2 SQL)	New columns: • PAGENUM • PQTY • STORTYPE • STORNAME • VCATNAME • FREEPAGE • PCTFREE • COMPRESS • COMPRESSRATIO • GBPCACHE • TRACKMOD • SECQTYI • PCTFREE_UPD • PCTFREE_UPD_CALC • INSERTALG	No
SYSTABLES catalog table (Db2 SQL)	New column: REGENERATETS. Changed column: TABLESTATUS.	No
SYSTABLESPACESTATS catalog table (Db2 SQL)	New columns: • GETPAGES • SYS_START • SYS_END • TRANS_START Changed column:	No
SYSSTATFEEDBACK catalog table (Db2 SQL)	Changed column: REASON.	No
SYSTRIGGERS catalog table (Db2 SQL)	New columns:	No

Catalog object	Change introduced	Incompatible change?
	• ACTIVE • ALTEREDTS • ASUTIME • DEBUG_MODE • ORIGINAL_CONTOK • REGENERATETS • SQLPL • VERSION • WLM_ENVIRONMENT • WRAPPED	
SYSUSERAUTH catalog table (Db2 SQL)	New columns: • SYS_START[1] on page 188 • SYS_END[1] on page 188 • TRANS_START[1] on page 188	No
SYSVARIABLES catalog table (Db2 SQL)	Changed columns: • DATATYPEID • SOURCETYPEID • DEFAULT • LENGTH • CCSID New index: DSNSNX06.	No
SYSVIEWDEP catalog table (Db2 SQL)	New columns: • SYS_START[1] on page 188 • SYS_END[1] on page 188 • TRANS_START[1] on page 188	No
SYSIBM.SYSTABSPACESTATS_H catalog table	New table. See Temporal versioning for Db2 catalog tables (Db2 SQL).	No
SYSAUDITPOLICIES_H catalog table	New unused table.	No
SYSCOLAUTH_H catalog table	New unused table.	No
SYSCONTEXT_H catalog table	New unused table.	No
SYSCONTEXTAUTHID_H catalog table	New unused table.	No
SYSCONTROLS_H catalog table	New unused table.	No
SYSCTXTTRUSTATTR_H catalog table	New unused table.	No
SYSDBAUTH_H catalog table	New unused table.	No
SYSPACKAUTH_H catalog table	New unused table.	No

Catalog object	Change introduced	Incompatible change?
SYSPLANAUTH_H catalog table	New unused table.	No
SYSRESAUTH_H catalog table	New unused table.	No
SYSROUTINEAUTH_H catalog table	New unused table.	No
SYSSCHEMAAUTH_H catalog table	New unused table.	No
SYSSEQUENCEAUTH_H catalog table	New unused table.	No
SYSTABAUTH_H catalog table	New unused table.	No
SYSUSERAUTH_H catalog table	New unused table.	No
SYSVIEWDEP_H catalog table	New unused table.	No
TEMPORAL_LOGICAL_TRANSACTION_TIME	New built-in global variable.	No
TEMPORAL_LOGICAL_TRANSACTIONS	New built-in global variable.	No

Notes:

1. Reserved for future IBM use.

GUPI

EXPLAIN table changes

The initial Db2 12 release introduces changes to the formats of certain EXPLAIN tables.

Important: Db2 12 introduces continuous delivery of new capabilities and enhancements in function levels. Most new capabilities become available only after activation of the Db2 12 function level that introduces them, or when applications run with the corresponding application compatibility level. For more information, see Chapter 12, "Activating Db2 12 function levels," on page 281.

PSPI

For the current formats, and sample CREATE TABLE statements for PLAN_TABLE and the other EXPLAIN tables, see member DSNTESC of the *prefix*.SDSNSAMP library. For the complete set of column descriptions for each table, see EXPLAIN tables (Db2 Performance) and Input tables (Db2 Performance).

Important: It is best to convert EXPLAIN tables to Db2 12 format during migration, or soon after migration. In Db2 12, the EXPLAIN function supports tables in Db2 12 or Db2 11 formats only. However, Db2 11 format EXPLAIN tables are deprecated. If you invoke EXPLAIN and Db2 11 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.

Result change for SQL statement EXPLAIN PACKAGE

When Db2 processes the SQL statement EXPLAIN PACKAGE, the HINT_USED column in the PLAN_TABLE is populated with EXPLAIN PACKAGE: *copy*. The *copy* field in the HINT_USED column will be one of the following values:

- "CURRENT" - the current copy
- "PREVIOUS" - the previous copy
- "ORIGINAL" - the original copy

DSNTIJG creates no EXPLAIN tables with SYSIBM qualifier

The DSNTIJSG installation job no long contains CREATE statements for EXPLAIN tables with the SYSIBM qualifier. Db2 no longer uses these tables.

New EXPLAIN tables

The initial Db2 12 release introduces no new EXPLAIN tables.

Changed EXPLAIN tables

The initial Db2 12 release introduces the following new and changed EXPLAIN table columns:

Table 9. List of new and changed EXPLAIN tables

EXPLAIN table	Changes

PLAN_TABLE (Db2 Performance)	**New columns**
	
	- PER_STMT_ID
	- AP_PLANID
	
	Changed columns
	
	- TNAME has a new value: 'DSN_SPIX_TBLX(*qblockno*)'.
	- COLUMN_FN_EVAL has new values: 'X' and 'Y'.
	- QBLOCK_TYPE has a new value: 'COTBLX'.
	- HINT_USED has:
	  - A new value: 'EXPLAIN SDQ: COPY *copy-id-number*'
	  - Changed values for *copy-id* in 'EXPLAIN PACKAGE: COPY *copy-id*'
	- PRIMARY_ACCESSTYPE has a new value: 'S'.

DSN_COLDIST_TABLE (Db2 Performance)	**New columns** - PER_STMT_ID - AP_PLANID
DSN_KEYTGTDIST_TABLE (Db2 Performance)	**New columns** - PER_STMT_ID - AP_PLANID
DSN_DETCOST_TABLE (Db2 Performance)	**New columns** - BLOCK_FETCH - PER_STMT_ID - AP_PLANID

Table 9. List of new and changed EXPLAIN tables (continued)

EXPLAIN table	Changes
DSN_FUNCTION_TABLE (Db2 Performance)	**New columns** • PER_STMT_ID • AP_PLANID
DSN_PGRANGE_TABLE (Db2 Performance)	**New columns** • PER_STMT_ID • AP_PLANID
DSN_PGROUP_TABLE (Db2 Performance)	**New columns** • PER_STMT_ID • AP_PLANID
DSN_PREDICAT_TABLE (Db2 Performance)	**New columns** • PER_STMT_ID • AP_PLANID
DSN_PREDICATE_SELECTIVITY table (Db2 Performance)	**New columns** • PER_STMT_ID
DSN_PTASK_TABLE (Db2 Performance)	**New columns** • PER_STMT_ID • AP_PLANID
DSN_SORT_TABLE (Db2 Performance)	**New columns** • PER_STMT_ID • AP_PLANID
DSN_SORTKEY_TABLE (Db2 Performance)	**New columns** • PER_STMT_ID
DSN_STAT_FEEDBACK (Db2 Performance)	**Changed columns** • REASON has a new value: 'STALE' • AP_PLANID

Table 9. List of new and changed EXPLAIN tables (continued)

EXPLAIN table	Changes
DSN_STATEMENT_CACHE_TABLE (Db2 Performance)	**New columns** • APPLCOMPAT • CNO • PER_STMT_ID • QUERY_HASH • QUERY_HASH_VERSION • STABILIZED • STAT_SUS_CHILDLLOCKS • STAT_SUS_OTHERLLOCKS • STAT_SUS_PPPLOCKS • STAT_SUS_PAGEPLOCKS • STAT_SUS_OTHERPLOCKS • STBLGRP • STAT_SUS_PIPE • STAT_SUS_PQSYNC • STMT_HASHID2 • STMT_HASH2VER • AP_PLANID • AP_PLANHASH • AP_PLANHASHVER • CONNECTION_TYPE • CLIENT_USERID • CLIENT_APPLNAME • CLIENT_WRKSTNNAME

Table 9. List of new and changed EXPLAIN tables (continued)

EXPLAIN table	Changes
DSN_STATEMNT_TABLE (Db2 Performance)	**New columns** • APCOMPARE_STATUS • APREUSE_STATUS • APREUSE_VERSION • APREUSE_COPYID • EXPLAIN_TYPE • FUNCTION_LVL • PER_STMT_ID • QUERY_HASH • STMT_HASHID2 • STMT_HASH2VER • AP_PLANID • AP_PLANHASH • AP_PLANHASHVER • AP_SERVICE_DATA • CONNECTION_TYPE • CLIENT_USERID • CLIENT_APPLNAME • CLIENT_WRKSTNNAME
DSN_STRUCT_TABLE (Db2 Performance)	**New columns** • PER_STMT_ID • AP_PLANID
DSN_VIEWREF_TABLE (Db2 Performance)	**New columns** • AP_PLANID
DSN_FILTER_TABLE (Db2 Performance)	**New columns** • AP_PLANID
DSN_QUERYINFO_TABLE (Db2 Performance)	**New columns** • AP_PLANID

PSPI

Related concepts
Investigating SQL performance by using EXPLAIN (Db2 Performance)
Input tables (Db2 Performance)
Related tasks
Migration step 25: Convert EXPLAIN tables to the current format (Db2 Installation and Migration)
Related reference
EXPLAIN (Db2 SQL)
EXPLAIN tables (Db2 Performance)

IFCID changes in Db2 12

Db2 12 introduces changes to various instrumentation facility component identifier (IFCID) trace fields.

Important: Db2 12 introduces continuous delivery of new capabilities and enhancements in function levels. Most new capabilities become available only after activation of the Db2 12 function level that introduces them, or when applications run with the corresponding application compatibility level. For more information, see Chapter 12, "Activating Db2 12 function levels," on page 281.

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.

Table 10. New IFCIDs

IFCID	Trace type and class	Mapping macro	Description
382	Accounting class 3 Accounting class 8 Monitor class 3 Monitor class 8 Accounting, Monitor	DSNDQW05	Records the beginning of a suspend for parallel task synchronization
383	Accounting class 3 Accounting class 8 Monitor class 3 Monitor class 8	DSNDQW05	Records the end of a suspend for parallel task synchronization
393	Performance class 3	DSNDQW05	Records information about phased-out package copies
389	Statistics class 8	DSNDQW05	Records information about indexes that have structures allocated for fast index traversal
411	Statistics class 10	DSNDQW05	Records detailed statistics about remote applications that connect to a local Db2 subsystem, based on the CURRENT CLIENT_APPLNAME special register
412	Statistics class 11	DSNDQW05	Records statistics about the client user IDs that are associated with remote connections to a local Db2 subsystem, based on the CURRENT CLIENT_USERID special register
413	Accounting class 3 Accounting class 8 Monitor class 3 Monitor class 8	DSNDQW05	Records the beginning of a wait for a pipe for an INSERT statement that is using insert algorithm 2

Table 10. New IFCIDs (continued)

IFCID	Trace type and class	Mapping macro	Description
414	Accounting class 3 Accounting class 8 Monitor class 3 Monitor class 8	DSNDQW05	Records the end of a wait for a pipe for an INSERT statement that is using insert algorithm 2
477	Performance class 4	DSNDQW05	Records allocation and deallocation of structures for fast index traversal

◁ PSPI

Changes to selected trace records

◁ PSPI ▷

The following table gives an overview of changes to specific IFCIDs. Changes to IFCID 0106, the system parameters record, and changes to serviceability trace records are not included.

Table 11. Changed IFCIDs

Enhancement	IFCID	Description of changes
Adaptive index processing	2, 125	Fields are added to record: • RID list processing statistics for adaptive index processing • Details about adaptive the index processing that was performed
Autonomous procedures	3	Fields are added to record accounting information for autonomous procedures.
Db2 exploitation of asynchronous cross-invalidation for coupling facility cache structures (APAR PH05193)	2	Fields are added to record: • The number of IXLCACHE (cache services) requests with asynchronous cross-invalidation • The number of IXLAXISN (coupling facility asynchronous cross-invalidation service) calls • The number of suspensions of the IXLAXISN service that occurred during waits for asynchronous cross-invalidation operations to complete.
	3	A field is added to record the number of suspensions of the IXLAXISN service that occurred during waits for cross-invalidation operations to complete.
Dynamic SQL plan stability	2, 21, 29, 30, 316	Fields are added to record: • The number of requests and satisfied requests for stabilized dynamic SQL statements • Locking for dynamic plan stability operations • Start and end of an EDM request for the SYSIBM.SYSDYNQRY table • Information on stabilized dynamic SQL statements in the dynamic statement cache

Table 11. Changed IFCIDs (continued)

Enhancement	IFCID	Description of changes
Fast index traversal	2	Fields are added to record information about when fast index traversal is used.
Improved detection and removal of incomplete connections (APAR PH44270)	1	Field QLSTCNVT is updated to include the number of connections to a server that were terminated as incomplete.
Information on conditional lock failures on a partition (APAR PH29098)	2	Fields are added to record: • The number of conditional lock failures on a partition during insert operations • The number of unconditional lock retries that occurred after a conditional lock request on a partition failed These counters can be used to diagnose issues like these in applications with insert operations: • Insert operations encounter a partition full message (reason code 00C9009C), but the table space is not full. • Insert operations fail with a lock timeout, but the table space is not full. • Insert operations have slow performance.
Integrated Accelerator for Z Sort use for GROUP BY or ORDER BY (APAR PH31684)	2, 96	Fields are added to record information about the use of the Integrated Accelerator for Z Sort coprocessor by Db2 sort operations for GROUP BY or ORDER BY.
Insert algorithm 2	2, 3, 18, 58, 239, 316, 401	Fields are added to record: • Wait times for pipes for insert algorithm 2 INSERT statements • Information about the use of pipes for insert algorithm 2 INSERT statements
Enhanced monitoring support for insert algorithm 2 (APAR PI81731)	2, 3, 148	Fields are added to record: • The number times that insert algorithm 2 was used. • The number of times that the basic insert algorithm (insert algorithm 1) is used. • The number attempts and successful attempts to re-enable pipes for insert algorithm 2 since Db2 restart • Whether query parallelism was disabled by the resource limit facility for at least one dynamic SELECT statement in a thread.
IBM zHyperLink exploitation for database synchronous read I/O (APAR PI87072) and active log write I/O (APAR PH05030)	1	Fields are added to record: • The number of eligible active log write I/O requests with successful IBM zHyperLink processing. • The number of eligible active log write I/O requests with unsuccessful IBM zHyperLink processing.

Table 11. Changed IFCIDs (continued)

Enhancement	IFCID	Description of changes
	2	Fields are added to record:
		• The number of successful synchronous read I/O operations that were performed using IBM zHyperLink processing.
		• The number of synchronous read I/O operations that resulted in disk cache hits, and did not use IBM zHyperLink processing.
	3, 148	Fields are added to record:
		• Wait time for synchronous read I/O operations that resulted in disk cache hits
		• The number of synchronous read I/O operations that resulted in disk cache hits.
	3, 148, 239	Fields are added to record:
		• The number of synchronous read I/O operations that used IBM zHyperlink processing.
		• The number of synchronous read I/O operations that resulted in disk cache hits.
		• The amount of CPU time for successful synchronous read I/O operations that used IBM zHyperlink processing.
	6	A flag is added to indicate whether a synchronous read I/O request occurred with or without IBM zHyperLink processing.
	7	A flag is added to indicate whether a synchronous read I/O request with IBM zHyperLink processing was successful.
	199	Fields are added to record the average and maximum delay and total number of pages read with synchronous read I/O and IBM zHyperLink processing.
Improved IFCID 0402 reset behavior (PH50729)	402	When the RESET_IFCID402 option is specified for the DDF_COMPATIBILITY subsystem parameter, 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:
		• QW0402TE
		• QW0402TQ
		• QW0402TF
		• QW0402TW
		• QW0402CE
		• QW0402CW
		• QW0402OE
		• QW0402OW
Lift partition limits	6, 7, 21, 44, 124, 127, 0128, 150, 172, 196, 198, 223, 226, 227, 255, 0259, 305	Fields are added to:
		• Indicate whether pages in range-partitioned table spaces use absolute or relative page numbering
		• Hold relative page numbers

Table 11. Changed IFCIDs (continued)

Enhancement	IFCID	Description of changes
Location statistics enhancements (APAR PH16111)	1, 365	Fields are added to record additional statistics that describe the behavior of connections between distributed locations and a local Db2 server. See PH16111 for details.
Processing improvements for data sets that are used by utilities (PH27493)	2	Fields are added to record: • The number of infrequently accessed data sets that were physically closed • The number of utility-access-only data sets that were physically closed
Real-time monitoring of CICS attachment facility calls (PH31447)	3	Fields are added to record CICS origin adapter data.
Rebind phase-in for packages that are being used for execution (APAR PH05989)	239	A field is added to record the ID for the current package copy.
Serviceability improvements to provide more information for insert operations that fail with reason codes 00C90090 or 00C9009C (PH29098)	2	Fields are added to record: • The number of failed conditional partition lock requests during insert operations • The number of times that a conditional partition lock request failed, and the lock request was retried with an unconditional lock request
Statement-level monitoring for accelerated Hybrid Transactional and Analytical Processing (HTAP) support (APAR PH00574, PH15534)	3, 316, 401	Fields are added to record: • The amount of time in milliseconds that statements waited for execution on an accelerator because the statements needed changes that were not yet applied to a referenced table. • The number of times that statement execution did not complete because changes were not applied to a referenced table before the WAITFORDATA delay protocol time period expired.

Table 11. Changed IFCIDs (continued)

Enhancement	IFCID	Description of changes
Statistics and accounting enhancements for incremental updates to IBM Db2 Analytics Accelerator that are processed using the Integrated Synchronization method (APAR PH18334 and PH26681)	2	Fields are added to record: • Accumulated CPU time, since Db2 startup, that was consumed on standard processors by all asynchronous log reader tasks. • Accumulated CPU time, since Db2 startup, that was consumed on an IBM z Integrated Information Processor (zIIP) by all asynchronous log reader tasks. • Accumulated CPU time, since Db2 startup, that was consumed on standard processors by all asynchronous log reader tasks, and those tasks were eligible for execution on a zIIP. • Accumulated CPU time, not including CPU time on a zIIP, that was consumed by all asynchronous log reader tasks that ran on behalf of an accelerator. • Accumulated CPU time on a zIIP that was consumed by all asynchronous log reader tasks that ran on behalf of an accelerator. • Accumulated CPU time that was consumed on standard processors by all asynchronous log reader tasks that ran on behalf of an accelerator, and those tasks were eligible for execution on a zIIP.
	3	A new product identifier, LRT, is added to identify threads that connect to Db2 to retrieve log records that were collected by asynchronous log reader tasks.
Statistics class 1 includes IFCID 0369 (APAR PH43916)	369	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.
System profile monitoring profile search improvements (APAR PH12041)	402	Fields are added to record: • The current number of active threads • The current number of suspended threads • The highest number of threads since DDF start • The current number of connections • The highest number of connections since DDF start
Transfer ownership	2, 62, 140, 361	Fields are added to record use of the TRANSFER OWNERSHIP statement.
Miscellaneous changes	2	Fields are added to record: • Details about buffer pool overflow • The number of times that RID list processing was not used. Fields that are related to the EDM statement pool are deleted.

Table 11. Changed IFCIDs (continued)

Enhancement	IFCID	Description of changes
	2	Changes to the maximum number of repeated sections in certain IFCID 2 records (PH55153): The following changes are made: • 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, 3	Fields are added to record: • Additional IRLM system information
	53	A field is added to record the section number of the SQL statement.
	58	A field is added to record the section number of the SQL statement.
	199	Fields are added to record I/O for read and castout, in microseconds.
	316	Fields are added to record: • Accumulated wait time due to global contention for L-locks and P-locks • Wait times that are at the same level of granularity as the wait times in class 3 accounting data
	401	Fields are added to record accumulated wait time due to global contention for L-locks and P-locks.
	All	A field is added to the correlation header with the batch job step name, to correlate an accounting record with a given job step.

Deleted trace records

IFCID 0366 is no longer supported.

An additional location for the IFCID flat file (DSNWMSGS)

You can now find DSNWMSGS in either of the following locations:

• The most current version of DSNWMSGS is available only for clients who have Db2 12 for z/OS licenses. The information is in a PDF file. To locate this information, see Db2 12 for z/OS IFCID flat file (DSNWMSGS).

• 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

Subsystem parameter changes in Db2 12

You can use this information to plan for subsystem parameter changes in Db2 12.

Important: Db2 12 introduces continuous delivery of new capabilities and enhancements in function levels. Most new capabilities become available only after activation of the Db2 12 function level that introduces them, or when applications run with the corresponding application compatibility level. For more information, see Chapter 12, "Activating Db2 12 function levels," on page 281.

Subsystem parameter changes in function level 501 and higher

The following changes take effect when you activate function level 501 or higher in Db2 12.

Function level	Subsystem parameter	Change introduced	Incompatible change?
FL 508	FTB NON UNIQUE INDEX field (FTB_NON_UNIQUE_INDEX subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter	No
FL 504	TS COMPRESSION TYPE field (TS_COMPRESSION_TYPE subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
FL 502	ENCRYPTION_KEYLABEL field in macro DSN6SPRM (Db2 Installation and Migration)	New subsystem parameter.	No

Subsystem parameter changes in function level 500

The following changes take effect when you activate function level 500 in Db2 12.

Subsystem parameter	Change introduced	Incompatible change?
AUTH_COMPATIBILITY in macro DSN6SPRM (Db2 Installation and Migration)	New subsystem parameter. * This new subsystem parameter is not an incompatible change. However, its purpose is to provide time to resolve specific authorization-related incompatible changes, and the default value specifies that Db2 12 uses the new authorization behavior. See "Changes to the authorization check of the UNLOAD utility" in "Security release incompatibilities" on page 140.	No *
CACHE DYN STABILITY field (CACHEDYN_STABILIZATION subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No

Subsystem parameter	Change introduced	Incompatible change?
COMPRESS DB2 DIR LOBS field (COMPRESS_DIRLOB subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
DDL MATERIALIZATION field (DDL_MATERIALIZATION subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
DEFAULT INSERT ALGORITHM field (DEFAULT_INSERT_ALGORITHM subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
PAGE SET PAGE NUMBERING field (PAGESET_PAGENUM subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
REORG IC LIMIT DASD field (REORG_IC_LIMIT_DASD subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
REORG IC LIMIT TAPE field (REORG_IC_LIMIT_TAPE subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
UTILS BLOCK FOR CDC field (UTILS_BLOCK_FOR_CDC subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
REORG INDEX NOSYSUT1 (REORG_INDEX_NOSYSUT1 subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	
SUBSTR COMPATIBILITY field (SUBSTR_COMPATIBILITY subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No

Subsystem parameter changes in function level 100

The following changes take effect when you migrate to function level 100 in Db2 12.

Subsystem parameter	Change introduced	Incompatible change?
AUTO BIND field (ABIND subsystem parameter) (Db2 Installation and Migration)	COEXIST behavior is used even when YES is specified. That is, no automatic bind occurs in release coexistence or fallback scenarios when a plan or package that was automatically bound in Db2 11 runs again in Db2 12. This change prevents repeating automatic remigration binds each time that a plan or package runs again on a different release during coexistence or fallback scenarios. For more information, see Automatic binds in coexistence (Db2 Installation and Migration).	No
ALLOW_UPD_DEL_INS_WITH_UR in macro DSN6SPRM (Db2 Installation and Migration)	New subsystem parameter.	No
ALTERNATE COPY POOL field (ALTERNATE_CP subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
APPL COMPAT LEVEL field (APPLCOMPAT subsystem parameter) (Db2 Installation and Migration)	The default value for new installations is changed to 'V12R1M500'. Support is also added for values that correspond to Db2 12 function levels.	No
CDDS_MODE in macro DSN6LOGP (Db2 Installation and Migration)	New subsystem parameter.	No
CDDS_PREFIX in macro DSN6LOGP (Db2 Installation and Migration)	New subsystem parameter.	No
MINUTES/CHECKPOINT field (CHKFREQ and CHKMINS subsystem parameters) (Db2 Installation and Migration)	The default value is changed to 3 and the accepted range for the MINUTES/CHECKPOINT value is now 1 - 5 minutes. If the CHKTYPE subsystem parameter setting is SINGLE and the CHKFREQ subsystem parameter setting is 6 - 60, Db2 uses 5 for the CHKFREQ setting, issues a warning message, and sets RC=4 for subsystem parameter assembly. If the CHKTYPE subsystem parameter setting is BOTH and the CHKMINS subsystem parameter setting is 6 - 1439, Db2 uses 5 for the CHKMINS setting, issues a warning message, and sets RC=4 for subsystem parameter assembly.	No
COPY FAST REPLICATION field (COPY_FASTREPLICATION subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
DDF_COMPATIBILITY in macro DSN6FAC (Db2 Installation and Migration)	A new value is supported: RESET_IFCID402 specifies that Db2 resets IFCID 0402 counters when the associated profile threshold changes.	No

Subsystem parameter	Change introduced	Incompatible change?
DEFAULT INSERT ALGORITHM field (DEFAULT_INSERT_ALGORITHM subsystem parameter) (Db2 Installation and Migration)	• The default value is changed from 2 to 1. • A new value is added, 0, which means that the basic insert algorithm (insert algorithm 1) is used regardless of the specification at the object level.	No
DISALLOW_SEL_INTO_UNION in macro DSN6SPRM (Db2 Installation and Migration)	The default is changed from NO to YES.	No
DISALLOW_SSARAUTH in macro DSN6SPRM (Db2 Installation and Migration)	New subsystem parameter.	No
EDM SKELETON POOL SIZE field (EDM_SKELETON_POOL subsystem parameter) (Db2 Installation and Migration)	The default value is changed to 51 200.	No
FLASHCOPY XRCP field (FLASHCOPY_XRCP subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
HSM MESSAGE DS HLQ field (UTILS_HSM_MSGDS_HLQ subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
INDEX MEMORY CONTROL field (INDEX_MEMORY_CONTROL subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter. (SELECTED,AUTO) and (SELECTED,n) settings are added to support enabling fast index traversal only for specific indexes. These settings are used when you specify YES in the FOR SELECTED INDEXES ONLY field on panel DSNTIP71.	No
INLISTP in macro DSN6SPRM (Db2 Installation and Migration)	The default value is changed to 1000.	No
LA_SINGLESEL_ISOCS_CDY subsystem parameter (Db2 Installation and Migration)	New subsystem parameter.	No
LOAD FORMAT DELIMITED IMPLICIT DECIMAL SCALE (LOAD_DEL_IMPLICIT_SCALE subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
LOAD RO OBJECTS field (LOAD_RO_OBJECTS subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
MFA AUTH UNUSED TIME field (MFA_AUTHCACHE_UNUSED_TIME subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No

Subsystem parameter	Change introduced	Incompatible change?
RID POOL SIZE field (MAXRBLK subsystem parameter) (Db2 Installation and Migration)	The default value is changed to 1000000 (1,000,000 kilobytes).	No
MAX UDFS field (MAX_UDF subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	Yes
NPGTHRSH in macro DSN6SPRM (Db2 Installation and Migration)	The default value is changed to 1, which means that Db2 favors matching index access, through the index with the most matching columns, for tables or partitions that have fewer than one data page. (For SAP installations, the default value remains 10.)	No
RO SWITCH TIME field (PCLOSET subsystem parameter) (Db2 Installation and Migration)	The default value is changed to 45.	No
PEER RECOVERY field (PEER_RECOVERY subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
PRIMARY QUANTITY field (PRIQTY subsystem parameter) (Db2 Installation and Migration)	The default value is changed to 125 (in cylinders—ACLUNIT is removed).	No
PROFILE AUTOSTART field (PROFILE_AUTOSTART subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
MANAGE REAL STORAGE field (REALSTORAGE_MANAGEMENT subsystem parameter) (Db2 Installation and Migration)	APAR PH47163 improves the behavior when the default setting AUTO is used, and introduces a new value AUTO1, which can be used to revert to the previous AUTO behavior if necessary. For more information, see APAR PH47163.	No
REORG TS NOPAD DEFAULT (REORG_TS_NOPAD_DEFAULT subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
REORG PART SORT NPSI field (REORG_PART_SORT_NPSI subsystem parameter) (Db2 Installation and Migration)	The behavior of the YES setting is changed so that Db2 always sorts the keys for non-partitioned secondary indexes (NPSIs). To continue with the previous behavior, use AUTO.	Yes
RLF SCOPE field (RLFENABLE subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
REMOTE STATIC SQL field (RLFERRDSTC subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No

Subsystem parameter	Change introduced	Incompatible change?
STATIC SQL field (RLFERRSTC subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
STATISTICS COLGROUP DATA SORT STG LIMIT field (STATCLGSRT subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
STATISTICS PROFILE FEEDBACK field (STATFDBK_PROFILE subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
DDF STATS TIME field (STATIME_DDF subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
MAIN STATS TIME field (STATIME_MAIN subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter. APAR PH56228 removes the control of the interval of location statistics trace records, IFCIDs 365, 411, and 412 from STATIME_MAIN.	No
PAGE-LEVEL SAMPLING field (STATPGSAMP subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
PREVENT_NEW_IXCTRL_PART in macro DSN6SPRM (Db2 Installation and Migration)	The default value is changed to YES.	No
SECONDARY QTY field (SECQTY subsystem parameter) (Db2 Installation and Migration)	The default value is changed to 15 (in cylinders—ACLUNIT is removed).	No
SMF ACCOUNTING field (SMFACCT subsystem parameter) (Db2 Installation and Migration)	The default value is changed to '1,2,3,7,8'.	No
DB BACKUP STG GROUP field (UTIL_DBBSG subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
LOG BACKUP STG GRP field (UTIL_LGBSG subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
UTILS USE ZSORT (UTILS_USE_ZSORT subsystem parameter) (Db2 Installation and Migration)	New subsystem parameter.	No
DB2 zHyperLinks SCOPE field (ZHYPERLINK subsystem	New subsystem parameter.	No

Subsystem parameter	Change introduced	Incompatible change?
parameter) (Db2 Installation and Migration)		
ACCELERATION OPTIONS field (QUERY_ACCEL_OPTIONS subsystem parameter) (Db2 Installation and Migration)	New field on panel DSNTIP8A: 13 ENABLE ACCELERATOR SPECIFIC RESULTS	No

Removed subsystem parameters in Db2 12 (function level 100 and higher)

The following tables lists subsystem parameters that are removed from this version of Db2 for z/OS. Refer to the information for the earlier version for detailed descriptions of the removed subsystem parameters.

Table 12. Removed subsystem parameters in Db2 12

Subsystem parameter	Setting used in Db2 12	Incompatible change?
ALCUNIT	CYL, which means that primary and secondary space allocations are always obtained in cylinder units.	No
CACHEPAC	10M, which is the previous maximum value.	No
CACHERAC	10M, which is the previous maximum value.	No
CACHE_DEP_TRACK_STOR_LIM	None, because the storage is allocated from the SCB pool.	No
CACHEDYN_FREELOCAL	0, which means that Db2 does not free cached dynamic statements to relieve high use of storage by dynamic SQL caching.	No
CATALOG (in DSN6ARVP)	YES, which means that Db2 requires all archive log data sets to be cataloged, regardless of whether they are allocated on disk or tape.	No
CHECK_SETCHKP	NO, which means that he CHECK DATA and CHECK LOB utilities do not place objects in CHECK PENDING status if an inconsistency is detected.	No
COMPRESS_SPT01	YES, which means that Db2 compresses the SPT01 table space during the next reorganization of the table space.	No
CONTSTOR	NO, which means that Db2 does not periodically contract the working storage area for each thread.	No
DB2SORT	ENABLE, which means that Db2 utilities use the Db2 Sort for z/OS product when it is installed. If Db2 Sort for z/OS is not installed, DFSORT is used instead.	No
EDMPOOL	None, because all EDM storage is above the bar.	No
EDPROP	NO (or DPROP SUPPORT field = 1 on the panel DSNTIPO), which means that Db2 always allows changes to tables defined with DATA CAPTURE CHANGES, but the SQL changes are not propagated.	No
CHGDC	NO (or DPROP SUPPORT field = 1 on the panel DSNTIPO), which means that Db2 always allows changes to tables defined with DATA CAPTURE CHANGES, but the SQL changes are not propagated.	No

Table 12. Removed subsystem parameters in Db2 12 (continued)

Subsystem parameter	Setting used in Db2 12	Incompatible change?
EDPROP	NO (DPROP SUPPORT field = 1 on the panel DSNTIPO), which means that Db2 always allows changes to tables defined with DATA CAPTURE CHANGES, but the SQL changes are not propagated.	No
CHGDC	NO (DPROP SUPPORT field = 1 on the panel DSNTIPO), which means that Db2 always allows changes to tables defined with DATA CAPTURE CHANGES, but the SQL changes are not propagated.	No
INDEX_IO_PARALLELISM	YES, which means that I/O parallelism is enabled.	No
IRLMAUT	Yes, which means that Db2 starts and stops the IRLM automatically.	No
IRLMSWT	120, which was the previous default value and specifies that Db2 abends if the IRLM does not start automatically within two minutes.	No
LEMAX	None. Db2 does not require an LE token limit for any package that was bound in DB2 10 or later. Db2 12 supports only packages that are bound in DB2 10 or later.	No
LOBVALA	Db2 automatically determines the amount of storage needed for processing LOB values.	No
LOBVALS	Db2 automatically determines the amount of storage needed for processing LOB values.	No
MGEXTSZ	YES, which specifies that Db2 automatically optimizes the secondary extent allocations of data sets for table spaces and index spaces that have a SECQTY value of greater than zero.	No
MINSTOR	NO, which means that Db2 does not use storage management algorithms that minimize the amount of working storage that is consumed by individual threads	No
NEWFUN *dsnhdecp* parameter	None, because the Db2 precompiler and coprocessor no longer use the NEWFUN *dsnhdecp* parameter.	No
OBJECT_CREATE_FORMAT	EXTENDED, which means that new indexes and table spaces are created to use the extended 10-byte RBA and LRSN log record format.	<u>Yes</u>
PCLOSEN	32767, which is the previous maximum value and specifies that Db2 converts a page set or partition from read-write to read-only after this number of consecutive log checkpoints.	No
REORG_IGNORE_FREESPACE	NO, which means that the REORG TABLESPACE utility uses the specified PCTFREE and FREEPAGE values.	No

Table 12. Removed subsystem parameters in Db2 12 (continued)

Subsystem parameter	Setting used in Db2 12	Incompatible change?
RRF	ENABLE, which means that newly created table spaces or newly added partitions on partition-by-growth table spaces are created in the reordered row format. Existing basic row format table spaces are converted to reordered row format by the LOAD REPLACE or REORG TABLESPACE utilities.	No
SPT01_INLINE_LENGTH	32138, which is the maximum size of LOB column data that is maintained in base tables in the SPT01 directory table space.	Yes
SQWIDSC	YES, which means that the database manager invalidates but does not quiesce SQL statements in the dynamic statement cache that depend on an object that is the target of a RENAME TABLE or DROP ALIAS statement.	No
UTILITY_OBJECT_CONVERSION	NOBASIC, which means that utilities convert existing table spaces and indexes that use 6-byte page format are converted to extended 10-byte page format. Table spaces and indexes that already use extended 10-byte page format cannot be returned to the 6-byte page format. **Important:** When the NOBASIC setting of the UTILITY_OBJECT_CONVERSION subsystem parameter is in effect, any LOAD, REORG, and REBUILD INDEX utility job that specifies RBALRSN_CONVERSION BASIC always fails. Similarly, if the target object is still in the 6-byte page format, these utility jobs also fail if they specify RBALRSN_CONVERSION NONE. Plan to remove the RBALRSN_CONVERSION option from these utility jobs before applying the PTF for PH26317.	Yes
UTSORTAL	YES, which means that the utility attempts to allocate the sort work data sets. To increase the possible degree of parallelism, the utility minimizes the number of sort work data sets that are used. If the SORTNUM clause is specified in the utility control statement and the IGNSORTN subsystem parameter is set to NO, the sort program is used to allocate the sort work data sets. The utility also does not allocate the work data sets if they are allocated in the JCL.	No
XMLVALA	Db2 automatically determines the amount of storage for processing XML values.	No
XMLVALS	Db2 automatically determines the amount of storage for processing XML values.	No

> GUPI

Related reference

Directory of subsystem parameters, panel fields, and application default values (Db2 Installation and Migration)

The extended 10-byte RBA and LRSN in Db2 12

Beginning in Db2 11, Db2 uses the extended format 10-byte RBA and LRSN values, and conversion of the BSDS to the extended format is required at migration to Db2 12.

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	**00000000**112233445566

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	**00**112233445566**000000**

The 3 bytes on the right side might be might be zero or x'FF', depending on the situation. For the beginning of an LRSN range, zeros are used. For the end of an LRSN range, x'FF' is used.

Internally, the values that are kept in memory are all 10 bytes, except when they need to be externalized to structures that remain in the 6-byte format. The values are stored internally as 10 bytes even in conversion mode. The conversion from the 10-byte values to 6-byte format is done at end points, such as when a log record is written, or when the PGLOGRBA field in a data or index page is updated.

When 10-byte format values are externalized

Extended RBA and LRSN values are externalized in the following contexts before objects are converted:

Messages

In Db2 12, 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 SYSLGRNX table (Db2 SQL).

Bootstrap data sets (BSDS)

The BSDS contains the LRSN and RBA values that bound each active and archive log data and a number of others that have various purposes.

⚠️ **Attention:** In Db2 subsystems that are not data sharing members, if Db2 is already at risk of reaching the 6-byte RBA limit, it is strongly recommended that you first convert all catalog and directory objects, then convert all user objects to the 10-byte RBA format, before migrating to Db2 12.

Before you can convert the BSDS, the Db2 subsystem or data sharing member must be started in Db2 11 new-function mode.

In data sharing, you must convert the BSDS for each Db2 member separately.

At migration to Db2 12, you cannot start Db2 12 until the BSDS is converted to use the 10-byte RBA and LRSN formats. You can convert the BSDS before or during the Db2 12 migration process.

⚠️ **Attention:** After the BSDS is converted to the 10-byte format, Db2 stops issuing messages to warn you about the risk of reaching the 6-byte RBA or LRSN limits. The increased size of all log records also accelerates progress toward the 6-byte RBA logging limit.

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).

For instructions, see Convert BSDS records to the extended 10-byte format: DSNTIJCB (Db2 Installation and Migration).

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 12, 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, before migrating to Db2 12.

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 00100000002FFFFFF is used because the TBasic table uses a 6-byte format LRSN.
- For the last update **F**, the only requirement is that the LRSN must be greater than the existing value, so the time of the log record is used.

The log entry for the fifth update **E** occurs later in the log for M10 (a higher RBA value) because it was delayed in generating the LRSN value. This situation requires that the last two updates are from different transactions. Otherwise, the sixth update must wait for the fifth update to complete to ensure that the sixth transaction has a later LRSN and a later sequence in the log.

Related 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)

Activation details summary for function levels

Use this information to plan activation of Db2 12 function levels. Activation of a function level implies activation of all lower function levels.

Before you activate function levels, you must also resolve any incompatible changes that are introduced by the activated function levels. For a summary of such changes, see "Incompatible changes summary for function levels 501 and higher" on page 119.

Activation details for function level 510 (V12R1M510)

Function level 510 (V12R1M510) has the following activation details:

Enabling APAR:	PH33727
Minimum Db2 catalog level:	V12R1M509
Catalog changes:	None.
Application compatibility control:	None.

Activation details for function level 509

The following activation details apply to function level 509:

Enabling APAR:	PH33015
Minimum Db2 catalog level:	V12R1M509
Catalog changes:	A new COMPRESS_USED column is added in the SYSIBM.SYSTABLEPART catalog table, and new values 'F', and 'H' are added for the COMPRESS column in the following catalog tables:

	• SYSIBM.SYSTABLESPACE
	• SYSIBM.SYSTABLEPART
Application compatibility control:	The following new capabilities require that application runs at application compatibility level V12R1M509 or higher:
	• UPDATE or DELETE from the parent table in a temporal RI relationship.

Activation details for function level 508

Function level 508 has the following activation details:

Enabling APAR:	APAR PH29392
Minimum Db2 catalog level:	V12R1M507
Catalog changes:	None
Application compatibility control:	Application compatibility level V12R1M508 or higher must be in effect for use of the new MOVE TABLE keyword of the ALTER TABLESPACE statement.

Activation details for function level 507

Function level 507 has the following activation details:

Enabling APAR:	PH24371.
Minimum Db2 catalog level:	V12R1M507
Catalog changes:	The following built-in global variables are introduced via a CATMAINT job:
	• MAX_LOCKS_PER_TABLE_SPACE
	• MAX_LOCKS_PER_USER
Application compatibility control:	Applications must run at application compatibility level V12R1M507 or higher to use the following new capabilities:
	• The SYSIBMADM.MAX_LOCKS_PER_TABLESPACE built-in global variable for overriding the NUMLKTS subsystem parameter.
	• The SYSIBMADM.MAX_LOCKS_PER_USER built-in global variable for overriding the NUMLKUS subsystem parameter.
	• CREATE PROCEDURE with the OR REPLACE and SPECIFIC clauses for external procedures and native SQL procedures.
	• Newly supported passthrough-only expressions.

Activation details for function level 506

Function level 506 has the following activation details:

Enabling APAR:	PH16829.
Minimum Db2 catalog level:	V12R1M505
Catalog changes:	None.

Application compatibility control:	Applications must run at application compatibility level V12R1M506 or higher to use the following new capabilities: • Newly supported names for existing built-in functions. • DROP statement enhancements for implicitly dropping explicitly created table spaces.

Activation details for function level 505

Function level 505 has the following activation details:

Enabling APAR:	PH09191.
Minimum Db2 catalog level:	V12R1M505 [1] on page 87
Catalog level V12R1M505 changes:	A new COPYID column is added in the following tables: • SYSIBM.SYSPACKDEP • SYSIBM.SYSPACKSTMT
Application compatibility control:	Applications must run at application compatibility level V12R1M505 or higher to use the following new capabilities: • All built-in functions described in "New built-in functions for encryption and decryption with key labels" on page 86. • Creating a new index with a DECFLOAT column as a key, or adding a DECFLOAT column to an existing index. • Defining a primary or unique constraint with a DECFLOAT column. • Temporal and archive transparency support for WHEN clause on triggers. .

Notes:

1. Before tailoring the Db2 catalog for catalog level V12R1M505, apply the PTFs for the following APARs: PH19720 and PH15258.

Activation details for function level 504

Function level 504 has the following activation details:

Enabling APAR:	PH07672.
Minimum Db2 catalog level:	V12R1M503
Application compatibility control:	Applications must run at application compatibility level V12R1M504 or higher to use the following new capabilities: • All new SQL syntax described in "New SQL syntax alternatives for special registers and NULL predicates" on page 91. • All built-in functions described in "Newly supported built-in functions with IBM Db2 Analytics Accelerator" on page 91. • Preventing the creation of deprecated object types, as described in "Prevent creation of new deprecated objects" on page 89.

Activation details for function level 503

Function level 503 has the following activation details:

Enabling APAR:	PH00506
Minimum Db2 catalog level:	V12R1M503
Catalog level V12R1M503 changes:	The REPLICATION_OVERRIDE built-in global variable is added. For a summary of changes in Db2 12, see "Catalog changes in Db2 12" on page 179.
Application compatibility control:	Applications must run at application compatibility level V12R1M503 or higher to use the following new capabilities: • The SYSIBMADM.REPLICATION_OVERRIDE built-in global variable for replication of system-period temporal tables and generated expression columns. • The temporal query result change for system-period temporal tables defined with the ON DELETE ADD EXTRA ROW attribute that also contain a DATA CHANGE OPERATION column.

Activation details for function level 502

Function level 502 has the following activation details:

Enabling APAR:	PI95511
Minimum Db2 catalog level:	V12R1M502
Catalog level V12R1M502 changes:	The KEYLABEL column is added in the following catalog tables: • SYSIBM.SYSINDEXES • SYSIBM.SYSSTOGROUP • SYSIBM.SYSTABLESPACE • SYSIBM.SYSTABLES **Restriction:** Before you can run CATMAINT to tailor the Db2 catalog for function level 502 or higher, you must first activate function level 500 or 501. That is, the DISPLAY GROUP command output must indicate `HIGHEST ACTIVATED FUNCTION LEVEL (V12R1M500)` or higher. This restriction prevents tailoring the Db2 catalog for function levels higher than 500 while fallback to Db2 11 remains possible. However, later activating a lower function level such as function level 100* does not restrict the CATMAINT operation. For a summary of changes in Db2 12, see "Catalog changes in Db2 12" on page 179.
Application compatibility control:	Applications must run at application compatibility level V12R1M502 or higher to use the following new capabilities: • CREATE or ALTER statements that specify the KEYLABEL option. However, applications that already reference encrypted data can continue to run at their current application compatibility levels. • Explicit casting of numeric data types to fixed or variable length graphic strings, with the GRAPHIC or VARGRAPHIC built-in functions or the CAST specification.

Function level 501 activation details

Function level 501 has the following activation details:

Enabling APAR:	PI70535
Minimum required catalog level:	V12R1M500
Application compatibility control:	Use of the LISTAGG built-in function level requires application compatibility level V12R1M501 or higher.
	IBM data server clients and drivers that use the capabilities in function level 501 or are bound with application compatibility level V12R1M501 or higher, require extra program preparation steps. For instructions, see Setting application compatibility levels for data server clients and drivers (Db2 Application programming and SQL).

Related tasks

Adopting new capabilities in Db2 12 continuous delivery
In Db2 12, function levels and application compatibility levels control the adoption of most new capabilities by Db2 subsystems and Db2 applications.

Function that Db2 12 no longer supports

If you are migrating to Db2 12 from Db2 11, be aware of the function that Db2 12 no longer supports.

The following functions are not supported in Db2 12.

Creating non-UTS table spaces

FL 504 Non-UTS table spaces for base tables are deprecated. CREATE TABLESPACE statements that run at application compatibility level V12R1M504 or higher always create a partition-by-growth or partition-by-range table space, and CREATE TABLE statements that specify a non-UTS table space (including existing multi-table segmented table spaces) return an error. However, you can use a lower application compatibility level to create table spaces of the deprecated types if needed, such as for recovery situations. For instructions, see Creating non-UTS table spaces (deprecated) (Db2 Administration Guide).

Creating hash-organized tables

FL 504 Hash-organized tables are deprecated. Beginning in Db2 12, packages that are bound with APPLCOMPAT(V12R1M504) or higher cannot create hash-organized tables or alter existing tables to use hash-organization. Existing hash-organized tables remain supported, but they are likely to be unsupported in the future.

CREATE SYNONYM statements

FL 504 In V12R1M504 application compatibility and higher, Db2 no longer supports CREATE SYNONYM statements. Create aliases with CREATE ALIAS statements instead of synonyms.

The DSNACCOR stored procedure

FL 504 The DSNACCOR stored procedure is no longer created or configured during the Db2 installation and migration process. Use the DSNACCOX stored procedure instead. See DSNACCOX stored procedure (Db2 SQL).

Subsystem parameters that are removed in Db2 12

The following table lists the subsystem parameters that are removed and the resulting behavior in Db2 12. For a summary of all subsystem parameter changes in Db2 12 including new and changed subsystem parameters, see "Subsystem parameter changes in Db2 12" on page 200.

Table 13. Removed subsystem parameters in Db2 12

Subsystem parameter	Setting used in Db2 12	Incompatible change?
ALCUNIT	CYL, which means that primary and secondary space allocations are always obtained in cylinder units.	No
CACHEPAC	10M, which is the previous maximum value.	No
CACHERAC	10M, which is the previous maximum value.	No
CACHE_DEP_TRACK_STOR_LIM	None, because the storage is allocated from the SCB pool.	No
CACHEDYN_FREELOCAL	0, which means that Db2 does not free cached dynamic statements to relieve high use of storage by dynamic SQL caching.	No
CATALOG (in DSN6ARVP)	YES, which means that Db2 requires all archive log data sets to be cataloged, regardless of whether they are allocated on disk or tape.	No
CHECK_SETCHKP	NO, which means that he CHECK DATA and CHECK LOB utilities do not place objects in CHECK PENDING status if an inconsistency is detected.	No
COMPRESS_SPT01	YES, which means that Db2 compresses the SPT01 table space during the next reorganization of the table space.	No
CONTSTOR	NO, which means that Db2 does not periodically contract the working storage area for each thread.	No
DB2SORT	ENABLE, which means that Db2 utilities use the Db2 Sort for z/OS product when it is installed. If Db2 Sort for z/OS is not installed, DFSORT is used instead.	No
EDMPOOL	None, because all EDM storage is above the bar.	No
EDPROP	NO (or DPROP SUPPORT field = 1 on the panel DSNTIPO), which means that Db2 always allows changes to tables defined with DATA CAPTURE CHANGES, but the SQL changes are not propagated.	No
CHGDC	NO (or DPROP SUPPORT field = 1 on the panel DSNTIPO), which means that Db2 always allows changes to tables defined with DATA CAPTURE CHANGES, but the SQL changes are not propagated.	No
EDPROP	NO (DPROP SUPPORT field = 1 on the panel DSNTIPO), which means that Db2 always allows changes to tables defined with DATA CAPTURE CHANGES, but the SQL changes are not propagated.	No
CHGDC	NO (DPROP SUPPORT field = 1 on the panel DSNTIPO), which means that Db2 always allows changes to tables defined with DATA CAPTURE CHANGES, but the SQL changes are not propagated.	No

Table 13. Removed subsystem parameters in Db2 12 (continued)

Subsystem parameter	Setting used in Db2 12	Incompatible change?
INDEX_IO_PARALLELISM	YES, which means that I/O parallelism is enabled.	No
IRLMAUT	Yes, which means that Db2 starts and stops the IRLM automatically.	No
IRLMSWT	120, which was the previous default value and specifies that Db2 abends if the IRLM does not start automatically within two minutes.	No
LEMAX	None. Db2 does not require an LE token limit for any package that was bound in DB2 10 or later. Db2 12 supports only packages that are bound in DB2 10 or later.	No
LOBVALA	Db2 automatically determines the amount of storage needed for processing LOB values.	No
LOBVALS	Db2 automatically determines the amount of storage needed for processing LOB values.	No
MGEXTSZ	YES, which specifies that Db2 automatically optimizes the secondary extent allocations of data sets for table spaces and index spaces that have a SECQTY value of greater than zero.	No
MINSTOR	NO, which means that Db2 does not use storage management algorithms that minimize the amount of working storage that is consumed by individual threads	No
NEWFUN *dsnhdecp* parameter	None, because the Db2 precompiler and coprocessor no longer use the NEWFUN *dsnhdecp* parameter.	No
OBJECT_CREATE_FORMAT	EXTENDED, which means that new indexes and table spaces are created to use the extended 10-byte RBA and LRSN log record format.	Yes
PCLOSEN	32767, which is the previous maximum value and specifies that Db2 converts a page set or partition from read-write to read-only after this number of consecutive log checkpoints.	No
REORG_IGNORE_FREESPACE	NO, which means that the REORG TABLESPACE utility uses the specified PCTFREE and FREEPAGE values.	No
RRF	ENABLE, which means that newly created table spaces or newly added partitions on partition-by-growth table spaces are created in the reordered row format. Existing basic row format table spaces are converted to reordered row format by the LOAD REPLACE or REORG TABLESPACE utilities.	No
SPT01_INLINE_LENGTH	32138, which is the maximum size of LOB column data that is maintained in base tables in the SPT01 directory table space.	Yes

Table 13. Removed subsystem parameters in Db2 12 (continued)

Subsystem parameter	Setting used in Db2 12	Incompatible change?
SQWIDSC	YES, which means that the database manager invalidates but does not quiesce SQL statements in the dynamic statement cache that depend on an object that is the target of a RENAME TABLE or DROP ALIAS statement.	No
UTILITY_OBJECT _CONVERSION	NOBASIC, which means that utilities convert existing table spaces and indexes that use 6-byte page format are converted to extended 10-byte page format. Table spaces and indexes that already use extended 10-byte page format cannot be returned to the 6-byte page format. **Important:** When the NOBASIC setting of the UTILITY_OBJECT_CONVERSION subsystem parameter is in effect, any LOAD, REORG, and REBUILD INDEX utility job that specifies RBALRSN_CONVERSION BASIC always fails. Similarly, if the target object is still in the 6-byte page format, these utility jobs also fail if they specify RBALRSN_CONVERSION NONE. Plan to remove the RBALRSN_CONVERSION option from these utility jobs before applying the PTF for PH26317.	Yes
UTSORTAL	YES, which means that the utility attempts to allocate the sort work data sets. To increase the possible degree of parallelism, the utility minimizes the number of sort work data sets that are used. If the SORTNUM clause is specified in the utility control statement and the IGNSORTN subsystem parameter is set to NO, the sort program is used to allocate the sort work data sets. The utility also does not allocate the work data sets if they are allocated in the JCL.	No
XMLVALA	Db2 automatically determines the amount of storage for processing XML values.	No
XMLVALS	Db2 automatically determines the amount of storage for processing XML values.	No

Query I/O parallelism

Query I/O parallelism is removed in Db2 12. Applications that used query I/O parallelism in Db2 11 are downgraded to use sequential access at run time. You can rebind the packages to enable Db2 to consider the use of CP parallelism instead in Db2 12.

DSNRLST*xx* table formats from before DB2 version 8

Starting in Db2 12, DSNRLST*xx* table formats and related index formats from before DB2 version 8 format are not supported. When Db2 detects DSNRLST*xx* tables with unsupported formats, it issues message DSNT731I.

For more information, see Resource limit facility tables (Db2 Performance) and Convert RLF tables to the current format (Db2 Installation and Migration).

Trace types

IFCID 0366 is no longer supported. Run a trace for IFCID 0376 to track information about SQL statements that have potential incompatible changes when you switch to a new application behavior.

JDBC 3.0 version of the IBM Data Server Driver for JDBC and SQLJ

Starting with APAR PH17211, the IBM Data Server Driver for JDBC and SQLJ that is based on the JDBC 3.0 specification is no longer supported. You need to upgrade to the driver that is based on the JDBC 4.0 specification. See Upgrading from the IBM Data Server Driver for JDBC and SQLJ with JDBC 3.0 support to the IBM Data Server Driver for JDBC and SQLJ with JDBC 4.0 support for more information.

Deprecated function in Db2 12

Certain features and capabilities that Db2 12 for z/OS supports are *deprecated*, meaning that their use is discouraged. Although they remain supported except as noted below in Db2 12, 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.

For a detailed list of deprecated function in Db2 12, see "Deprecated function in Db2 12" on page 220.

Deprecated function in Db2 12

Certain features and capabilities that Db2 12 for z/OS supports are *deprecated*, meaning that their use is discouraged. Although they remain supported except as noted below in Db2 12, 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 14. Deprecated features and capabilities in Db2 12

Deprecated function	Recommended alternative	Support removed
AUTHCACH subsystem parameter	Use the default value.	Db2 13
Basic row format table spaces	Use reordered row format. Starting in Db2 12, any table space that uses basic row format is automatically converted to reordered row format when you run one of the following utilities: • LOAD REPLACE with the ROWFORMAT RRF option, or LOAD REPLACE without the ROWFORMAT option. The ROWFORMAT option is deprecated and will be removed eventually. • REORG TABLESPACE with the ROWFORMAT RRF option, or REORG TABLESPACE without the ROWFORMAT option. The ROWFORMAT option is deprecated and will be removed eventually.	—
BIND PLAN command MEMBER option	Use BIND PACKAGE commands to bind DBRMs into packages explicitly.	—

Table 14. Deprecated features and capabilities in Db2 12 (continued)

Deprecated function	Recommended alternative	Support removed
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 KEEPDYNAMIC refresh for DBATs (Db2 Performance) and Enabling high-performance DBATs (Db2 Performance).	—
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.	—
DDF_COMPATIBILITY subsystem parameter	Use the default value NULL.	Db2 13
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.	—
DSNRLMT*xx* table format from before Db2 11	Use the current format for resource limit facility (RLF) tables. See Convert RLF tables to the current format (Db2 Installation and Migration).	—
SYSPROC.DSNTBIND stored procedure	Use the SYSPROC.ADMIN_COMMAND_DSN stored procedure.	—
SYSPROC.DSNUTILS stored procedure	Use the SYSPROC.DSNUTILU stored procedure.	—

Table 14. Deprecated features and capabilities in Db2 12 (continued)

Deprecated function	Recommended alternative	Support removed
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.	—
DSVCI subsystem parameter	Use the default value YES.	Db2 13
EXTRAREQ subsystem parameter	Use the default value 100.	Db2 13
EXTRASERV subsystem parameter	Use the default value 100.	Db2 13
External SQL procedures	Use native SQL procedures, which are more fully supported, easier to maintain, and typically perform better that external SQL procedures. For more information, see Migrating an external SQL procedure to a native SQL procedure (Db2 Application programming and SQL)	—
Hash-organized tables	Alter tables to drop hash organization, and , create indexes to support fast index traversal in Db2 12 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.
HONOR_KEEPDICTIONARY subsystem parameter	Use the default value NO.	Db2 13
IMMEDWRI subsystem parameter	Use the default value NO.	Db2 13
LOAD utility IDENTITYOVERRIDE option	Use the OVERRIDE(IDENTITY) option.	—
LOAD utility PERIODOVERRIDE option	Use the OVERRIDE(SYSTEMPERIOD) option.	—
LOAD utility TRANSIDOVERRIDE option	Use the OVERRIDE(TRANSID) option.	—
MATERIALIZE_NODET_SQLTUDF subsystem parameter	Set MATERIALIZE_NODET_SQLTUDF to YES. Later Db2 releases always use the behavior of this setting.	—
MAXARCH subsystem parameter	Use the default value 10000.	Db2 13
MAXTYPE1 subsystem parameter	Use the default value 100.	Db2 13

Table 14. Deprecated features and capabilities in Db2 12 (continued)

Deprecated function	Recommended alternative	Support removed
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).	—
OPT1ROWBLOCKSORT subsystem parameter	Use the default value DISABLE.	Db2 13
PARA_EFF subsystem parameter	Use the default value 50.	Db2 13
PassTickets for RACF-protected user IDs.	Use client certificate authentication.	—
PLANMGMTSCOPE subsystem parameter	Use the default value STATIC.	Db2 13
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.	—
RESYNC subsystem parameter	Use the default value 2.	Db2 13
RBALRSN_CONVERSION keyword of the LOAD, REORG, and REBUILD INDEX utilities	Starting in Db2 12 with APAR PH26317, utilities that specify RBALRSN_CONVERSION(BASIC) always fail. If the target object is in the 6-byte page format, utilities also fail if they specify RBALRSN_CONVERSION(NONE)	Db2 12
REPAIR VERSIONS utility	Use the REPAIR CATALOG utility.	—

Table 14. Deprecated features and capabilities in Db2 12 (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.SOAPHTTPC supplied user-defined function	Use the DB2XML.SOAPHTTPNC user-defined function.	—
DB2XML.SOAPHTTPV supplied user-defined function	Use the DB2XML.SOAPHTTPNV user-defined function.	—
SUBQ_MIDX subsystem parameter	Use the default value ENABLE.	Db2 13
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.	Use partition-by-growth or partition-by-range universal table spaces instead. FL 504 In Db2 12, Packages bound with APPLCOMPAT(V12R1M504) or higher cannot create objects of the following types: • New partitioned (non-UTS) table spaces • New segmented (non-UTS) table spaces • New tables in existing segmented (non-UTS) table spaces • New tables in existing simple table spaces You cannot create new simple table spaces in any supported Db2 release.	—
TRACSTR subsystem parameter	Use the default value NO.	Db2 13

Related tasks
Preparing your system to install or migrate to Db2 12 (Db2 Installation and Migration)

Summary of operational requisite changes for Db2 function levels

Db2 12 function levels introduce changes to the operational requisites for Db2 12 for z/OS. *Operational requisites* are products that are required and must be present on the system or products that are not required but should be present on the system for Db2 to operate all or part of its functions.

Db2 function levels up to and including function level 510 introduce the following changes to the operational requisites for Db2 that are described in the Program Directory for Db2 12 for z/OS: No changes at this time.

Chapter 4. Application enablement in the Db2 12 initial release

The initial release of Db2 12 introduces new capabilities and enhancements for application enablement.

Important: Db2 12 introduces continuous delivery of new capabilities and enhancements in function levels. Most new capabilities become available only after activation of the Db2 12 function level that introduces them, or when applications run with the corresponding application compatibility level. For more information, see Chapter 12, "Activating Db2 12 function levels," on page 281.

Related information
IBM Db2 12 for z/OS Technical Overview (IBM Redbooks)

Advanced trigger support

Db2 12 introduces support for *advanced triggers*, while continuing to support the existing triggers created in previous releases, which are now considered *basic triggers*.

Any trigger created before activation of function level 500 or higher in Db2 12, or in earlier Db2 releases, is a *basic trigger*. Advanced triggers are supported at application compatibility level V12R1M500 or higher.

Advanced triggers offer the following advantages over basic triggers:

- In the trigger definition, an advanced trigger can:

  - Include more types of SQL statements, including SQL PL control statements, dynamic SQL statements, and SQL comments.
  - Define and reference more types of variables, including SQL variables and global variables.
  - Explicitly specify bind options.
  - Define multiple versions of the trigger.

- All transition variables are nullable.
- ALTER TRIGGER statements can change options, and change or regenerate the trigger body.
- The OR REPLACE clause can be used in CREATE TRIGGER (advanced) statements. It enables the use of a single CREATE statement to either define a new trigger or trigger version, or update an existing trigger or trigger version if it already exists.

For more information about the differences between basic and advanced triggers, see Triggers (Introduction to Db2 for z/OS).

Related tasks
Converting existing triggers to support advanced capabilities (Db2 Application programming and SQL)
Related reference
CREATE TRIGGER (advanced) (Db2 SQL)
CREATE TRIGGER (basic) (Db2 SQL)
ALTER TRIGGER (basic) (Db2 SQL)

Additional support for arrays

Db2 12 enhances array support in SQL statements.

Db2 for z/OS Db2 11 provided support for defining and manipulating arrays within SQL PL routines, and for passing arrays as parameters between routines.

Array support in Db2 12 adds the following enhancements:

- The ability to define a global variable as an array type. A global variable that is defined as an array type can be manipulated within an SQL PL context, or outside an SQL PL context.
- The following ARRAY_AGG function improvements:

  – ARRAY_AGG can be invoked without an ORDER BY clause.

  – ARRAY_AGG can be used with associative arrays.

Related concepts
Array types (Db2 SQL)
Global variables (Db2 SQL)
Related reference
ARRAY_AGG (Db2 SQL)
CREATE VARIABLE (Db2 SQL)

Additional support for global variables

Db2 12 enhances global variable support.

The following enhancements are included in Db2 12:

- Definition of a global variable with a BLOB, CLOB, or DBCLOB data type
- In a SET *assignment-statement*, specification of global variables as the target, and a *row-subselect* as the source.

Related reference
CREATE VARIABLE (Db2 SQL)

Additional support for pureXML®

Db2 12 enhances XML performance in several ways.

In Db2 12 of Db2 for z/OS, you can perform multiple document updates within a single XMLMODIFY invocation.

The updates can be in one of the following expressions in the *xquery-update-constant* of the XMLMODIFY function:

- A sequence expression
- A FLWOR expression that contains an updating expression in its return clause
- A conditional expression that contains an updating expression in its THEN or ELSE clause

In addition, Db2 12 improves the performance of XML queries by optimizing the choice of access plans.

The performance of XMLTable is also improved, in terms of the CPU time for constructing rows out of repeated elements that contain name-value pairs.

The XSLTRANSFORM function provides the ability to transform an XML document into a different data format. The output format can be any form possible for the XSLT processor, including, but not limited to, XML, HTML, and plain text.

Related reference
XMLMODIFY (Db2 SQL)

Additional support for JSON

Db2 12 adds support for the requirements imposed by web and mobile applications, with two ways of working with JSON: the Java driver for the JSON API or through SQL extensions.

The JSON_VAL function is enhanced in that the first argument is no longer required to be a BLOB column. In Db2 12 the first parameter can be:

- a view column
- a CASE expression
- a table expression with UNION ALL
- a trigger transition variable
- a SQL PL variable or parameter

MERGE statement enhancements

Db2 12 introduces changes to the SQL MERGE statement that provide more functionality and improve compatibility with the Db2 family.

Enhancements to the MERGE statement include support for:

- *table-reference* as an alternative way of specifying source data for the MERGE statement
- Multiple MATCHED clauses
- Additional predicates with MATCHED or NOT MATCHED
- DELETE operations
- IGNORE and the SIGNAL statement as actions

Related reference
MERGE (Db2 SQL)

SQL pagination support

Db2 12 introduces SQL syntax that enables dividing a query result table into separate pieces, so that, for example, mobile and web applications can present a subset of the rows of the result table at a time.

- With the growth of web and mobile applications, application developers are looking for more efficient ways to develop good-performing applications. Db2 12 introduces the OFFSET clause to define the number of rows to skip from the beginning of the query result table:

`GUPI`

```
SELECT * FROM TAB OFFSET 10 ROWS FETCH FIRST 10 ROWS ONLY;
```

`GUPI`

- Also available with Db2 12 is data-dependent pagination, which uses row value expressions in a basic predicate, which enables an application to access part of a Db2 result table based on a logical key value:

`GUPI`

```
WHERE (LASTNAME, FIRSTNAME) > ('SMITH', 'JOHN')
```

`GUPI`

Prior to Db2 12, such a statement had to be coded as:

`GUPI`

```
WHERE (LASTNAME = 'SMITH' AND FIRSTNAME > 'JOHN') OR (LASTNAME > 'SMITH')
```

`GUPI`

Related tasks
Accessing part of a result set based on data position (Db2 Application programming and SQL)
Related reference
offset-clause (Db2 SQL)

Basic predicate (Db2 SQL)

fetch-clause (Db2 SQL)

Support for Unicode columns in an EBCDIC table

Db2 12 extends support for Unicode columns in EBCDIC tables. A Unicode column in an EBCDIC table that is defined with Db2 12 is subject to fewer restrictions than if the column had been defined using Db2 11.

As businesses become more global, there is a need to store a wider set of characters. For example, with the expansion of EU countries, the customer name from a newly joined country might not be stored correctly in an SBCS EBCDIC column. Db2 12 addresses this need by providing a byte-based Unicode column of CCSID 1208 or CCSID 1200 in an EBCDIC table.

Unicode columns created in Db2 11 can be migrated to Db2 12 Unicode columns.

Related concepts
Unicode columns in EBCDIC tables (Db2 SQL)
Related tasks
Migrating Db2 11 Unicode columns in EBCDIC tables to Db2 12 or later Unicode columns in EBCDIC tables (Db2 SQL)

Piece-wise deletion of data

Db2 12 introduces functionality that helps to mitigate the effects of locking, when potentially millions of rows could be affected by a DELETE statement.

Db2 12 adds FETCH FIRST *n* ROWS ONLY syntax to the searched DELETE statement to limit the number of rows that are deleted in a single SQL statement.

Related reference
DELETE (Db2 SQL)
fetch-clause (Db2 SQL)

Support for temporal referential constraint

Db2 12 introduces referential constraints for application-period temporal tables.

In previous versions, temporal referential integrity could only be enforced using triggers or stored procedures. In Db2 12, a temporal referential constraint can be defined for an application-period temporal table that contains a BUSINESS_TIME period. You can specify the PERIOD BUSINESS_TIME clause in the definition of a referential constraint that enforces a temporal referential constraint for an application-period temporal table.

FL 509 At application compatibility level V12R1M509 or higher, temporal reference integrity is enhanced to remove a restriction against UPDATE or DELETE statements on the parent table.

Related concepts
Temporal tables and data versioning (Db2 Administration Guide)
Related tasks
Creating an application-period temporal table (Db2 Administration Guide)

More flexibility in defining application periods for temporal tables

In Db2 12, you can create an application-period temporal table with a BUSINESS_TIME period that is inclusive-inclusive. In previous versions, only inclusive-exclusive periods were supported.

For application-period temporal tables, an application period indicates the period of time when a row is valid. This period is defined by a begin column and an end column. In previous versions of Db2, the begin value of a period was always inclusive, but the end value of a period was always exclusive. This

type of period is called an inclusive-exclusive period. For example, if the begin column has a value of '01/01/2015' and the end column has a value of '03/21/2015', the row is valid from 01/01/2015 to 3/20/2015, including both of those dates. The row is not valid on 03/21/2015.

Beginning in Db2 12, you can define an application period to be inclusive-inclusive. For an inclusive-inclusive period, the end value is considered to be part of the period. To specify an inclusive-inclusive period, use the new INCLUSIVE keyword in the PERIOD BUSINESS_TIME clause of the CREATE TABLE or ALTER TABLE statement. By default, an application period is inclusive-exclusive.

Whether you should use an inclusive-inclusive period or an inclusive-exclusive period depends on what you think is the most natural way to express a certain period of time. For example, for an insurance policy, an inclusive-inclusive period might make sense. A policy can be valid from July 1 to December 31, where both end points are part of the period. In contrast, an inclusive-exclusive period might make sense for hotel reservations. You check in on June 30 and check out on July 4. In this case, you do not want July 4 to be included in the nights that the hotel room is reserved. An inclusive-exclusive period might also be useful when the end value is the end of the month. You can specify a period from February 1 through March 1, not including March 1. By specifying the period this way, you do not have to figure out whether the last day of February is the 29th or the 28th.

Related concepts
Temporal tables and data versioning (Db2 Administration Guide)
Related tasks
Creating an application-period temporal table (Db2 Administration Guide)
Related reference
CREATE TABLE (Db2 SQL)
ALTER TABLE (Db2 SQL)

Support for temporal logical transactions

Db2 12 introduces temporal logical transaction support for system-period temporal tables (SYSTEM_TIME periods).

Temporal logical transaction support applies to SYSTEM_TIME periods and applications using Db2 system-period temporal tables.

Db2 now supports logical units of temporal work that are not determined by commit and rollback. The values for row-begin and row-end columns are determined by applications based on a built-in global variable that you set.

Related concepts
Temporal tables and data versioning (Db2 Administration Guide)

PERCENTILE function support

Db2 12 introduces the PERCENTILE_CONT and PERCENTILE_DISC functions, which compute a percentile value within a given group.

PERCENTILE_CONT treats the group as a continuous distribution, meaning that if the percentage falls between two values, the result is interpolated between those two values. For example, the median (PERCENTILE_CONT(0.5)) of a group of 6 numbers would be average of the 3rd and 4th number, because the interpolated middle value between those two is found.

PERCENTILE_DISC treats the group as a set of discrete values. The result is not interpolated and is always one of the values in the group. PERCENTILE_DISC(0.5) over a group of 6 numbers returns the 3rd number.

Related reference
PERCENTILE_CONT (Db2 SQL)
PERCENTILE_DISC (Db2 SQL)

DRDA fast load

Db2 12 introduces DRDA fast load, which enables quick and easy loading of data from files residing on distributed clients.

The Db2 Call Level Interface (CLI) APIs, JDBC APIs, and Command Line Processor (CLP) have been enhanced to support remote loading of data to Db2 for z/OS. This new feature is supported in all Db2 client packages.

For information on how to invoke DRDA fast load, see the following topics:

CLP ZLOAD command
Loading a Db2 for z/OS table by using an IBM Data Server Driver for JDBC and SQLJ method
Loading a Db2 for z/OS table by using a client CLI program (Db2 Administration Guide)

Before using DRDA fast load, you must bind the DSNUT121 package at each location from which you want to load data. The following example binds the DSNUT121 package at a remote location:

```
BIND PACKAGE(location.DSNUT121)
      MEMBER(DSNUGSQL) -
      ACTION(ADD) ISOLATION(CS) ENCODING(EBCDIC) -
      VALIDATE(BIND) CURRENTDATA(NO) -
      LIBRARY('prefix.SDSNDBRM')
```

ODBC enhancements

Db2 12 introduces enhancements in the Db2 ODBC driver for z/OS to improve performance and portability.

Enhancements to the ODBC driver include support for:

• New ODBC INI keyword KEEPDYNAMIC and connection attribute SQL_ATTR_KEEP_DYNAMIC

• Support for the TIMESTAMP WITH TIMEZONE data type

Related concepts
Programming for ODBC (Db2 Programming for ODBC)

Obfuscated source code for SQL routines and triggers

Db2 12 introduces a method for protecting the encapsulated logic in SQL stored procedures, SQL functions, and triggers. In *obfuscated data definition statements* for SQL procedures, SQL functions, and triggers, the SQL logic is rendered unreadable, enabling the delivery of SQL routines and triggers without sharing the intellectual property of the SQL PL logic.

The data definition statements for routines and triggers can execute in obfuscated form. In obfuscated statements, only the name of the routine or trigger, the parameters, and the WRAPPED keyword are readable. The rest of the statement is encoded in such a way that it is not readable but can be decoded by a database server that supports obfuscated statements.

You can use the WRAP built-in function to create the obfuscated form of the data definition statements for SQL procedures, SQL functions, routines, and triggers. The CREATE_WRAPPED stored procedure also supports the deployment of obfuscated routines and functions by calling the WRAP built-in function.

Important: The obfuscation algorithm is not strong encryption, and it is not meant for use in security contexts.

Related tasks
Obfuscating source code of SQL procedures, SQL functions, and triggers (Db2 Administration Guide)
Related reference
WRAP (Db2 SQL)
CREATE_WRAPPED stored procedure (Db2 SQL)
CREATE FUNCTION (compiled SQL scalar) (Db2 SQL)

CREATE FUNCTION (inlined SQL scalar) (Db2 SQL)

CREATE FUNCTION (SQL table) (Db2 SQL)

CREATE PROCEDURE (SQL - native) (Db2 SQL)

CREATE TRIGGER (basic) (Db2 SQL)

CREATE TRIGGER (advanced) (Db2 SQL)

Support for maintaining session data on the target server

Db2 12 introduces an enhancement to reduce network traffic through the use of session tokens on the Db2 server.

Prior to Db2 12, in order to support transaction rerouting and pooling, distributed clients needed session data (special registers and global variables) to be returned and replayed.

With Db2 12, session data is maintained on the Db2 server when clients interchange session tokens. This enhancement results in reduced network traffic.

Related reference

Built-in session variables (Db2 SQL)

Resource limits for static SQL statements

The resource limit facility can be used to limit CPU execution times and resource consumption by static SQL statements that are bound in packages.

In previous versions of Db2, limits specified in the resource limit facility applied only to dynamic SQL statements.

Db2 12 introduces the ability to specify reactive governing for static SQL statements by adding new rows in resource limit facility tables. Static SQL statements are governed by rows that specify the RLFFUNC='A' (for DSNRLST*xx* tables) and RLFFUNC='B' (for DSNRLMT*xx* tables) values in the resource limit tables.

You can use this function to cancel poorly performing SQL statements that consume too many resources, especially in high-volume OLTP environments.

The value of the RLFENABLE subsystem parameter controls whether resource limits apply to dynamic SQL statements only, static SQL statements only, or to all SQL statements.

You specify default resource limit actions for static SQL statements by setting the values of the RLFERRSTC (for local statements) and RLFERRDSTC (for remote statements) subsystem parameters.

Related tasks

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

Setting default resource limits for SQL statements (Db2 Performance)

Related reference

Resource limit facility tables (Db2 Performance)

RLF SCOPE field (RLFENABLE subsystem parameter) (Db2 Installation and Migration)

STATIC SQL field (RLFERRSTC subsystem parameter) (Db2 Installation and Migration)

REMOTE STATIC SQL field (RLFERRDSTC subsystem parameter) (Db2 Installation and Migration)

Db2 REST services improve efficiency and security

The Db2 REST services support, available in Db2 12, unleashes your enterprise data and applications on Db2 for z/OS for the API economy. Mobile and cloud application developers can use efficiently created, consumable, scalable, and RESTful services by using any REST client. They can use these services to securely interact with business-critical data and transactions, without special Db2 for z/OS expertise.

Recent service updates to the Db2 REST services support introduces the BIND SERVICE and FREE SERVICE commands and versioning to REST services. Db2 database administrators can issue these DSN commands to create and drop user-defined REST services the same way as they use BIND PACKAGE

and FREE PACKAGE for normal application package creation and deletion. The Db2 commands, START RESTSVC, STOP RESTSVC, and DISPLAY RESTSVC, were also introduced with the recent service updates to the Db2 REST services support. The commands can be used by DBAs to manage the availability of user-defined REST services.

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

Modern language support Db2 for z/OS application development

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

> IBM data server client and driver types
> IBM Data Server Driver Package (Windows)
> Programming for Db2 for z/OS (Introduction to Db2 for z/OS)

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

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

> IBMDb2 for z/OS Package (UrbanCode Deploy)
> UrbanCode Deploy Tour
> Deploy database applications againstDb2 for z/OS with IBM UrbanCode deploy

Related concepts
Db2 application provisioning, development, and deployment (DevOps) solutions (Db2 Installation and Migration)
Related information
UrbanCode Deploy

Chapter 5. Administrator function in the Db2 12 initial release

The initial release of Db2 12 introduces enhancements and new capabilities for administrators.

Important: Db2 12 introduces continuous delivery of new capabilities and enhancements in function levels. Most new capabilities become available only after activation of the Db2 12 function level that introduces them, or when applications run with the corresponding application compatibility level. For more information, see Chapter 12, "Activating Db2 12 function levels," on page 281.

Related information
IBM Db2 12 for z/OS Technical Overview (IBM Redbooks)

Improved reliability and contention relief for online migration

In Db2 12, you can identify applications that might be incompatible with catalog migrations and updates.

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

The new command and function are also available in Db2 11, and can be used for both system and non-system databases. For more information, see Identify applications that are incompatible with online migration (Db2 Installation and Migration).

Related reference
-DISPLAY BLOCKERS (Db2) (Db2 Commands)
BLOCKING_THREADS (Db2 SQL)

Single-phase catalog migration in Db2 12

The Db2 12 catalog migration is accomplished in a single phase.

Migration to the Db2 12 catalog is completed by running the DSNTIJTC job. The following migration modes from previous releases are eliminated in Db2 12:

- Conversion Mode (CM)
- Enabling-new-function mode (ENFM)
- New-function mode (NFM)
- * modes (including CM* and ENFM*)

In Db2 12, you can use Db2 function levels and application compatibility levels to control the adoption of new capabilities and enhancements . For details, see Part 2, "Adopting new capabilities in Db2 12 continuous delivery," on page 261 and Application compatibility levels in Db2 12 (Db2 Application programming and SQL).

Important: Do not issue the ACTIVATE command or run job DSNTIJAF for activation of Db2 12 until you are certain that the subsystem or data sharing group can proceed on Db2 12, without the possibility of falling back to or coexistence with Db2 11. In data sharing, the ACTIVATE command has group scope. Fallback and coexistence become impossible with the successful activation of function level 500 or higher.

Tip: GUPI 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 9, "Determining the Db2 code level, catalog level, and function level," on page 271 GUPI

Related concepts

What's new in Db2 12

Db2 12 for z/OS takes Db2 to a new level, both extending the core capabilities and empowering the future. Db2 12 extends the core with new enhancements to scalability, reliability, efficiency, security, and availability. Db2 12 also empowers the next wave of applications in the cloud, mobile, and analytics spaces.

Related tasks

Migrating your Db2 subsystem to Db2 12 (Db2 Installation and Migration)

Related reference

Db2 12 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.

Installation or migration without requiring SYSADM

In Db2 12, you can use the installation SYSOPR authority to install or migrate a Db2 subsystem. You are not required to use the SYSADM or installation SYSADM authority to perform installation or migration tasks. Using the installation SYSOPR authority prevents unintended access to user objects during installation or migration.

For installing or migrating Db2, the installation SYSOPR authority is enhanced to access catalog and directory objects, create, drop, or alter system objects, execute the -ACTIVATE command, specify the owner for the BIND and REBIND commands, and run the CATMAINT utility. However, you are not allowed any access to non-system objects or user data in the subsystem.

The overall process of installing or migrating Db2 with the installation SYSOPR authority is the same as that with the SYSADM authority. You can use the installation SYSOPR authority to perform all installation or migration tasks if you meet the following additional requirements:

- The primary or secondary authorization ID that performs installation or migration tasks must have the installation SYSOPR authority.
- The current SQLID that processes data definition statements must be set to SYSINSTL which owns all the system objects created by the installation SYSOPR authority.
- The OWNER option for the BIND and REBIND commands must be set to an ID that is authorized to bind and execute SQL statements in a package.
- The new DSNTIAIN plan must be used with the DSNTIAD program to process all dynamic data definition statements for Db2 installation or migration.
- An explicit qualifier must not be specified for the CREATE TABLE or CREATE INDEX statement. The qualifier can be implicitly specified through a leading SET CURRENT SCHEMA statement so that SYSINSTL can own the table or index.

Related concepts

Required authorization for installation or migration (Db2 Installation and Migration)

Related reference

DSNTIPG: Installation preferences panel (Db2 Installation and Migration)

Support for preserve mirror options for recovery

Db2 12 supports the preserve mirror options for the RESTORE SYSTEM and RECOVER utilities. These options are useful if you use Peer-to-Peer Remote Copy (PPRC) to mirror your Db2 production volumes to a remote site for high availability.

During recovery, you can use the preserve mirror options to override the DFSMShsm preserve mirror settings that were in effect when the system-level backup was created. To do so, use the new FLASHCOPY_PPRCP option of RESTORE SYSTEM or RECOVER.

In previous versions of Db2, if you had to override or change the preserve mirror options during recovery from a system-level backup, you had to invoke DFSMShsm directly.

Related concepts
Preserve Mirror FlashCopy (z/OS DFSMSdss Storage Administration)
Related reference
Syntax and options of the RECOVER control statement (Db2 Utilities)
Syntax and options of the RESTORE SYSTEM control statement (Db2 Utilities)
Related information
FlashCopy to PPRC primary (DFSMS Advanced Copy Services)

Improved availability when altering index compression

To improve the availability of indexes in universal table spaces, alterations to index compression are a pending change in Db2 12.

For indexes in universal table spaces, alterations to index compression are a pending change that place the index in advisory REORG-pending (AREOR) status. Applications can continue to access the indexes. The updated value for the COMPRESS attribute in the ALTER INDEX statement is materialized by a subsequent online REORG INDEX or online REORG TABLESPACE at the table space level. With this improvement, database administrators can correct or remove a pending change to index compression without affecting the target index. Also, this improvement reduces the planning and costs that are associated with an application outage caused by the previous behavior.

Note: If the index is defined with the DEFINE NO attribute and data sets are not created yet, the alteration is still immediate. Also, for an index that is not in a universal table space, an alteration to index compression can be a pending change if other pending changes exist at the index, table, or table space level when the ALTER INDEX COMPRESS statement runs.

In previous releases, any alteration to use index compression placed the index in REBUILD-pending (RBDP) status. This behavior prevented applications from using the index until the REORG TABLESPACE utility or the REBUILD INDEX utility completed.

Related tasks
Compressing indexes (Db2 Performance)
Related reference
ALTER INDEX (Db2 SQL)

Online schema enhancements

Db2 12 improves system and application availability by extending the scope of recovery to a point in time before materialization of pending definition changes. In addition, in Db2 12, column alterations can be pending definition changes.

Improvements in recovery to a point in time before materialization of pending definition changes

Db2 11 introduces the capability to recover the following objects to a point in time before the pending definition changes were materialized:

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

Db2 12 extends recovery to a point in time before pending definition changes were materialized in the following ways:

- The object that is being recovered can be a partition-by-growth table space.

Pending column alterations

In Db2 12, ALTER TABLE ALTER COLUMN statements that change the data type, length, precision, or scale of columns can be pending alterations. By default, column alteration operations are immediate operations. To make column alterations in a subsystem pending operations, set the DDL_MATERIALIZATION subsystem parameter to ALWAYS_PENDING.

Improved availability when altering index compression

To improve the availability of indexes in universal table spaces, alterations to index compression are a pending change in Db2 12. For indexes in universal table spaces, alterations to index compression are a pending change that place the index in advisory REORG-pending (AREOR) status. Applications can continue to access the indexes. The updated value for the COMPRESS attribute in the ALTER INDEX statement is materialized by a subsequent online REORG INDEX or online REORG TABLESPACE at the table space level. With this improvement, database administrators can correct or remove a pending change to index compression without affecting the target index. Also, this improvement reduces the planning and costs that are associated with an application outage caused by the previous behavior.

Note: If the index is defined with the DEFINE NO attribute and data sets are not created yet, the alteration is still immediate. Also, for an index that is not in a universal table space, an alteration to index compression can be a pending change if other pending changes exist at the index, table, or table space level when the ALTER INDEX COMPRESS statement runs.

In previous releases, any alteration to use index compression placed the index in REBUILD-pending (RBDP) status. This behavior prevented applications from using the index until the REORG TABLESPACE utility or the REBUILD INDEX utility completed.

Related concepts
Point-in-time recovery (Db2 Utilities)
Pending column alterations (Db2 Administration Guide)
Related tasks
Recovering to a point in time before pending definition changes were materialized (Db2 Administration Guide)
Related reference
DDL MATERIALIZATION field (DDL_MATERIALIZATION subsystem parameter) (Db2 Installation and Migration)

Improved catalog availability

Db2 12 improves the availability of Db2 catalog objects on which CATMAINT and online REORG run.

PSPI

When a transaction issues dynamic SQL statements, the Db2 database manager dynamically prepares the SQL statements for execution. During preparation of SQL statements, the database manager acquires read claims on several catalog table spaces and related indexes, and acquires a DBD lock on the catalog. The DBD lock is needed to serialize catalog operations with CATMAINT and other DDL that may execute against the catalog, because CATMAINT might be making structural changes to the catalog.

In previous releases of Db2, the transaction released the DBD lock and the read claims at commit points. If transactions with dynamic SQL statements did not issue commit operations for a long time, CATMAINT and online REORG on the catalog were blocked during that long period.

In Db2 12, the Db2 database manager releases DBD locks on the catalog and read claims against catalog objects as soon as PREPARE statement execution is complete. This change improves availability for CATMAINT and online REORG on catalog objects.

PSPI

Object ownership transfer

In Db2 12, you can transfer user-defined objects to another user or role by using the new TRANSFER OWNERSHIP statement. The new owner can be an authorization ID or a role.

The privilege set for the TRANSFER OWNERSHIP statement includes the ownership of the object or the SECADM authority. The SEPARATE_SECURITY system parameter does not impact the SECADM authority.

Related reference
TRANSFER OWNERSHIP (Db2 SQL)

Improved data validation after running DSN1COPY

After the DSN1COPY utility runs, mismatches between the data page format and the catalog description of the format can result. The REPAIR utility is enhanced to identify and fix more of those mismatches.

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

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

Beginning in Db2 12, the REPAIR CATALOG utility enables the following mismatches to be fixed:

- The column data type or length in the table space differs from the catalog definition for the column. If Db2 supports conversion from the data type and length in the table space to the data type and length in the column, REPAIR CATALOG enables conversion of the data type or length of the data in the table space to match the catalog definition the next time that the data is accessed.

In Db2 12, the REPAIR CATALOG TEST utility is enhanced to detect the following mismatches:

- The table space is a partition-by-range table space with absolute page numbering, but the catalog indicates that the table space has relative page numbering.
- The table space is partition-by-range with relative page numbering, but the catalog indicates that the table space has absolute page numbering.
- The table space is a partition-by-range table space with absolute page numbering, but the catalog indicates that the table space has relative page numbering. REPAIR CATALOG changes the catalog definition to match the table space format.
- The number of columns in the table space is greater than the number of columns in the catalog definition of the table.
- The column data type or length in the table space differs from the catalog definition for the column.

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

Related reference
DSN1COPY (Db2 Utilities)
Syntax and options of the REPAIR control statement (Db2 Utilities)

Alternate copy pools for system-level backups

In Db2 12, you can create extra system-level backups on alternate backup storage groups without affecting the primary system-level backup.

In previous releases, when you used the BACKUP SYSTEM utility to create system-level backups, you could use up to two copy pools only, one for the database and one for logs. These copy pools define the storage groups to copy and the backup storage groups to store the copies.

Beginning in Db2 12, you can also optionally use additional copy pools called alternate copy pools. An *alternate copy pool* includes the same defined set of storage groups as the standard copy pool; however different backup storage groups are specified. Therefore, you can make system-level backups on different target volumes. These target volumes can be shared among different Db2 subsystems. That is, a volume can be used by another subsystem if it is not part of an active copy. You can make these extra system-level backups without overwriting the primary production system-level backup or invalidating any versions in the regular backup storage group.

You can specify that you want to use an alternate copy pool when you make the system-level backup. To do so, specify the ALTERNATE_CP option and the related backup storage group options (DBBSG and LGBSG) on the BACKUP SYSTEM utility control statement.

Alternatively, you can specify that you want BACKUP SYSTEM to regularly alternate between using the standard copy pool and the alternate copy pool. For this behavior, use the ALTERNATE_CP subsystem parameter and the related backup storage group subsystem parameters (UTIL_DBBSG and UTIL_LGBSG).

The system-level backups that are made with the alternate copy pool can be used during recovery by the RESTORE SYSTEM utility or the RECOVER utility.

Related concepts
Copy pools (Db2 Utilities)
Related reference
Syntax and options of the BACKUP SYSTEM control statement (Db2 Utilities)
Syntax and options of the RESTORE SYSTEM control statement (Db2 Utilities)
Syntax and options of the RECOVER control statement (Db2 Utilities)
ALTERNATE COPY POOL field (ALTERNATE_CP subsystem parameter) (Db2 Installation and Migration)
DB BACKUP STG GROUP field (UTIL_DBBSG subsystem parameter) (Db2 Installation and Migration)
LOG BACKUP STG GRP field (UTIL_LGBSG subsystem parameter) (Db2 Installation and Migration)

New DSNUTILV stored procedure

In Db2 12, you can use the new DSNUTILV stored procedure to pass utility statements as large as 2 GB. It is otherwise similar to the existing DSNUTILU stored procedure, which cannot pass utility statements larger than 32 KB.

The DSNUTILV stored procedure has the following restrictions:

- Any keywords, names, or constants in the utility statement cannot be larger than 32,704 bytes.
- If you pass a file reference variable, the specified data set that contains the utility statements must be in variable record format.

For detailed information about the requirements, see the description of the *utstmt* input parameter in the DSNUTILV information.

The DSNUTILV stored procedure is created after new function is activated in Db2 12.

Related tasks
Installing and configuring Db2-supplied routines that use new function: DSNTIJRT (Db2 Installation and Migration)
Related reference
DSNUTILV stored procedure (Db2 SQL)

Automatic start of system profile tables at Db2 start

In Db2 12, you can specify that profiles take effect immediately when Db2 is started.

The PROFILE_AUTOSTART subsystem parameter specifies whether a START PROFILE command is issued automatically when Db2 is started.

Related tasks

Monitoring and controlling Db2 by using profile tables (Db2 Administration Guide)

Related reference

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

-START PROFILE (Db2) (Db2 Commands)

Increased partition sizes and simplified partition management for partition-by-range table spaces with relative page numbering

Db2 12 introduces a new attribute for partition-by-range table spaces featuring partition sizes up to 1 TB and less-disruptive partition management.

partition-by-range table spaces with relative page numbering allow the following improvements in space allocation:

- Data and index partition sizes can be up to 1 TB.
- Greater flexibility in growing your partitions. With relative page numbering, you can grow partitions by any number of gigabytes. With absolute page numbering partition growth is restricted to gigabytes in powers of two.
- Different partitions of the same table can have different size limits.
- Improved availability because DSSIZE can be increased for individual partitions as an immediate ALTER, without requiring a REORG.

You create table spaces with the new attribute by specifying the PAGENUM RELATIVE option of the CREATE TABLESPACE statement. Existing table spaces can be converted with the ALTER TABLESPACE statement.

Related concepts

Partition-by-range table spaces (Db2 Administration Guide)

Related tasks

Creating table spaces explicitly (Db2 Administration Guide)

Ability to add partitions between existing logical partitions

Db2 12 introduces the ability to add partitions between existing logical partitions in partition-by-range table spaces.

Adding a partition between existing logical partitions is supported only for partition-by-range (UTS) table spaces. The new partition is the next physical partition not being used until the maximum for the table space has been reached. A partition added between existing logical partitions is a pending definition change if the data sets are already defined. Otherwise, the change is immediate.

Related tasks

Adding partitions (Db2 Administration Guide)

Related reference

ALTER TABLE (Db2 SQL)

More efficient point-in-time recovery

In Db2 12, the default behavior for a RECOVER utility job with the TOLOGPOINT option or the TORBA option is to skip unnecessary recoveries. If an object has not been changed since the recovery point,

the utility does not recover that object. Therefore, RECOVER can avoid unnecessary work and use fewer resources. This change can also potentially reduce the time that objects are unavailable and improve the overall recovery time.

This new behavior is controlled by the SCOPE option. The default value is SCOPE UPDATE, which specifies that unchanged objects are to be skipped during recovery. However, you can explicitly specify SCOPE ALL to indicate that all objects are to be processed by the RECOVER utility, regardless of whether they have changed since the recovery point.

Related reference
Syntax and options of the RECOVER control statement (Db2 Utilities)

UNLOAD privilege for the UNLOAD utility

A new UNLOAD privilege is introduced Db2 12 to provide better control for executing the UNLOAD utility.

In Db2 12, the SELECT privilege is no longer sufficient to execute the UNLOAD utility.

Before the activation of function level 500 or higher, IFCID 404 trace can be used to identify the usage of the SELECT privilege for UNLOAD utility access. The UNLOAD privilege can be granted prior to activation of, but it does not become effective until new function is activated. When new function is activated, the UNLOAD privilege takes effect by default.

After new function activation, if you want the SELECT privilege to be checked for UNLOAD utility access, set the AUTH_COMPATIBILITY system parameter to SELECT_FOR_UNLOAD. For more information, see "Changes to the authorization check of the UNLOAD utility" in "Security release incompatibilities" on page 140.

Related tasks
Run premigration queries (DSNTIJPM) (Db2 Installation and Migration)
Related reference
AUTH_COMPATIBILITY in macro DSN6SPRM (Db2 Installation and Migration)
UNLOAD (Db2 Utilities)
GRANT (Db2 SQL)
REVOKE (Db2 SQL)

Temporal versioning for Db2 catalog tables

Db2 12 can use temporal versioning to record historical information for certain catalog tables. You can use the historical information in statistics-related catalog tables to analyze, predict, and help prevent specific conditions in a subsystem.

FL 500

Related concepts
Temporal tables and data versioning (Db2 Administration Guide)
Recovery of temporal tables with system-period data versioning (Db2 Administration Guide)
When Db2 externalizes real-time statistics (Db2 Performance)
Related tasks
Creating a system-period temporal table (Db2 Administration Guide)
Querying temporal tables (Db2 Administration Guide)
Collecting history statistics (Db2 Performance)

REORG enhancements

Db2 12 provides enhancements to the REORG utility.

Db2 12 introduces the following improvements to REORG:

Improved FlashCopy® management

A REORG with only a FlashCopy inline image copy causes the REORG to fail if the FlashCopy itself fails.

Improved partition-level partition-by-growth REORGs

Supports creation of a new PBG partition for overflow rows during a partition-level REORG.

Prevention of COPY-pending on a LOB table space during REORG of partition-by-growth table spaces

Avoids leaving COPY pending on LOB table spaces of new partition-by-growth partitions grown during the log phase of REORG.

REORG-level management of delete of PBG partitions

Adds a new keyword, DROP_PART, for REORG empty partition-by-growth partition pruning when ZPARM alteration is not feasible.

Support for the new COMPRESSRATIO catalog column

Adds to LOAD, REORG, and RUNSTATS the ability to gather average compression ratio at the record level instead of the page saved level.

Additional offload to zIIP

The RELOAD phase can now be offloaded to zIIP specialty engines.

New mapping table format support

New format supports 7-byte RIDs needed with partition-by-range table spaces with relative page numbering.

REORG against RO pagesets

You can run REORG with any SHRLEVEL against any RO table space and indexes.

Display of claimer information

You can display claimer information on each drain failure, not just last retry.

Related concepts

Statistics collection enhancements for SQL performance
Db2 12 provides several statistics collection enhancements, including several of which improve SQL performance.

Related reference

REORG TABLESPACE (Db2 Utilities)

LOAD enhancements

Db2 12 introduces enhancements to the LOAD utility.

Db2 12 introduces the following improvements to the LOAD utility:

LOAD PART REPLACE with dummy input against an empty partition-by-range (PBR) partition

You can now benefit from reduced elapsed and CPU time.

LOAD SHRLEVEL CHANGE PARALLEL support for partition-by-growth (PBG)

Removes the single input parallelism restriction for PBG table spaces for LOAD SHRLEVEL CHANGE.

Additional zIIP offload

You can offload the RELOAD phase of the LOAD utility, including the data conversion and loading of the record into the page set.

LOAD RESUME BACKOUT YES to avoid RECP on failure

Adds a new option on LOAD RESUME SHRLEVEL NONE to allow LOAD to back out the rows already loaded upon encountering an error (such as conversion, LOB/XML, duplicate key, referential integrity violation) without leaving the page set in RECP.

PREFORMAT support for auxiliary tables

Support is extended to LOB table spaces and auxiliary indexes.

Maintain MAXASSIGNEDVAL for identity columns

LOAD maintains the MAXASSIGNEDVAL for user-provided input.

LOAD REPLACE support for COMPRESSRATIO column
Gathers the average compression ratio at the record level instead of the page saved level.

Related concepts
Statistics collection enhancements for SQL performance
Db2 12 provides several statistics collection enhancements, including several of which improve SQL performance.

Automated statistics profiles
Db2 12 introduces the capability for Db2 to manage which statistics to collect based on the queries running on the system.

Related reference
LOAD (Db2 Utilities)

Backup and recovery enhancements

Db2 12 introduces enhancements to the BACKUP and RECOVERY utilities.

Db2 12 introduces the following improvements to the BACKUP and RECOVERY utilities:

Point-in-time (PIT) recovery support for PBG table spaces
You can recover PBG table spaces to a PIT prior to a REORG that materialized these physical pending alters: page size (buffer pool) DSSIZE, SEGSIZE, member cluster.

FLASHCOPY_PPRCP keyword
You can specify the FLASHCOPY_PPRCP keyword for the RESTORE SYSTEM and RECOVER utilities, to specify or override the preserve mirror option for PPRC (sync disk mirroring) on the utility statement.

New default behavior for the PIT RECOVER utility
Point-in-time recovery is skipped for non-updated pagesets, which saves resources (CPU) and time.

System Level Backup support for multiple copypools
You can keep extra system-level backups on disk during upgrades.

Improved messaging for BACKUP SYSTEM and RESTORE SYSTEM
DFSMShsm messages included in the utility job output for BACKUP SYSTEM and RESTORE SYSTEM provide improved messaging.

COPY option to specify FASTREPLICATION(REQUIRED)
You can control whether FlashCopy is required during creation of the FlashCopy image copy.

New MODIFY RECOVERY option DELETEDS
You can use this to delete image copy data sets.

New MODIFY RECOVERY option NOCOPYPEND
You can use this to specify that MODIFY RECOVERY not set COPY pending.

More accurate validation of recovery scenarios
When you execute MODIFY RECOVERY at the table space level a SYSCOPY record for each partition is inserted, which allows a more accurate validation of recovery scenarios.

Related tasks
Recovering with the BACKUP SYSTEM and RESTORE SYSTEM utilities (Db2 Administration Guide)

Statistics collection enhancements for SQL performance

Db2 12 provides several statistics collection enhancements, including several of which improve SQL performance.

RUNSTATS utility enhancements

Db2 12 introduces the following improvements to RUNSTATS:

Enhanced CLUSTERRATIO formula
Ensures that CLUSERRATIO better reflects dynamic prefetch behavior.

Specifying FREQVAL without the COUNT n keywords
If you specify FREQVAL without the COUNT n keywords, Db2 automatically determines the appropriate number to collect.

Autonomic statistics maintenance
In Db2 12, missing statistics can be automatically identified and included in the set of recommended statistics in the statistics profile. This enhancement frees you from having to manually manage RUNSTATS.

Column group statistics support
Column group statistics are supported with LOAD PARALLEL.

INVALIDATECACHE option to avoid dynamic cache invalidation
You can specify the INVALIDATECACHE keyword to prevent invalidating the dynamic cache (for example, LOAD, REBUILD INDEX, REORG INDEX, REORG TABLESPACE, and RUNSTATS) when collecting statistics and to avoid the unnecessary work of rebinds.

Inline statistics enhancements

Db2 12 introduces the following improvements to inline statistics. The utilities that provide inline statistics are LOAD, REBUILD INDEX, REORG INDEX, and REORG TABLESPACE :

PROFILE support for inline stats in REORG and LOAD
Profiles are now supported with inline statistics to match RUNSTATS functional support. This means that the latest PROFILE settings will be picked up so the appropriate statistics are gathered during LOAD and REORG (thus eliminating an additional execution of RUNSTATS).

Inline stats support for LOAD PARALLEL
You can now specify LOAD PARALLEL to provide inline statistics functional parity with RUNSTATS.

Keywords added to the REBUILD INDEX and REORG INDEX utilities
The MOST, LEAST, and BOTH keywords are added to the REBUILD INDEX and REORG INDEX utilities.

Keywords supported for indexes
The MOST, LEAST, and BOTH keywords are supported for indexes in the LOAD and REORG TABLESPACE utilities.

COLGROUP performance improvement
COLGROUP performance is improved when COLGROUP column specifications are redundant with INDEX columns. The duplicate COLGROUP is ignored in lieu of the index processing.

Related reference
RUNSTATS TABLESPACE syntax and options (Db2 Utilities)
RUNSTATS (Db2 Utilities)

Data sharing enhancements

Db2 12 introduces various data sharing enhancements including peer recovery, improved automatic LPL an GRECP recovery, and asynchronous coupling facility (CF) lock duplexing.

Db2 12 introduces the following data sharing enhancements:

- Db2 Data sharing members can use *peer recovery* to automatically recover retained locks for a failed member.
- If automatic LPL and GRECP recovery fails, Db2 automatically retries after three minutes, for a total of three retries.
- Asynchronous lock duplexing reduces the overhead of CF lock duplexing.

Related concepts
Peer recovery for data sharing members (Db2 Data Sharing Planning and Administration)
Db2 for z/OS support for asynchronous CF duplexing (Db2 Installation and Migration)
Related reference
Data sharing status descriptions (Db2 Data Sharing Planning and Administration)

Dynamic plan stability

Db2 12 introduces the capability to achieve access path stability comparable to static SQL statements for repeating cached dynamic SQL statements.

When you enable *dynamic SQL plan stability,* Db2 stores statement cache structures for specified dynamic SQL statements in the Db2 catalog. Whenever a *stabilized dynamic SQL statement* is not present in the dynamic statement cache when issued, Db2 can load the statement cache structures from the Db2 catalog and avoid the full prepare operation.

Related concepts
Dynamic SQL plan stability (Db2 Performance)
Related tasks
Stabilizing access paths for dynamic SQL statements (Db2 Performance)
Related reference
-START DYNQUERYCAPTURE (Db2) (Db2 Commands)

Automated statistics profiles

Db2 12 introduces the capability for Db2 to manage which statistics to collect based on the queries running on the system.

The Db2 cost-based optimizer relies on statistics about tables and indexes. Prior to Db2 12, it was often the case that only standard or default statistics were gathered, which meant that queries might not perform as well as possible. It was sometimes difficult to know which statistics should be gathered, due to the number of SQL statements, tables, indexes, columns, and keys.

Db2 12 introduces the automatic updating of statistics profiles based on recommendations generated during query optimization, so that RUNSTATS collects all of the statistics recommended by the optimizer.

FL 507 Beginning in function level 507, when statistics are collection with the RUNSTATS TABLESPACE, REORG TABLESPACE, or LOAD utilities with the USEPROFILE option, the utility also deletes any existing statistics that are not included in the profile.

Related concepts
Statistics profiles (Db2 Performance)
Related tasks
Applying statistics recommendations to statistics profiles automatically (Db2 Performance)
Identifying missing or conflicting statistics (Db2 Performance)
Related reference
The LOG option of the LOAD or REORG utilities (Db2 Administration Guide)

Real-time statistics improvements

Db2 12 introduces more information to aid users in diagnosing problems with real-time statistics collection, and in collecting more accurate statistics.

Those improvements are:

- New messages DSNT535I and DSNT536I, which identify unavailable resources that prevent externalization of real-time statistics.

- The ability to keep statistics history

- The addition of a column that tracks GETPAGEs

Related concepts
How real-time statistics are used by Db2 utilities (Db2 Utilities)
Automated statistics profiles

Db2 12 introduces the capability for Db2 to manage which statistics to collect based on the queries running on the system.

Support for compressing LOB data

Db2 12 supports compressing LOB data when zEnterprise® data compression (zEDC) hardware is available to the z/OS environment. By compressing your LOB data, you can reduce the size of data in a LOB table space to ensure that the LOB table space and its associated base table space remain available for inserting data.

You can enable LOB compression for individual table spaces by specifying the COMPRESS YES option in a CREATE LOB TABLESPACE or ALTER TABLESPACE statement.

LOB data compression can also be enabled on all LOB table spaces that are associated with the Db2 directory. To enable LOB compression for the Db2 directory, set the COMPRESS_DIRLOB subsystem parameter to YES.

LOB compression requirements

The following conditions are required for LOB compression in Db2 for z/OS:

- Db2 12 must be at function level 500 or higher.
- The zEDC hardware and software must be available and configured in the z/OS system. For more information, see Requirements for zEnterprise Data Compression.

  In data sharing, system performance can degrade dramatically if a member accesses compressed LOB data and the zEDC hardware and software are not configured in z/OS.
- The LOB table space must be associated with a base table that is in a universal table space.
- The total length of the entire LOB must be larger than the defined data page size, otherwise the LOB is not compressed.

Related reference
CREATE LOB TABLESPACE (Db2 SQL)
COMPRESS DB2 DIR LOBS field (COMPRESS_DIRLOB subsystem parameter) (Db2 Installation and Migration)
Overview and planning of zEnterprise Data Compression (zEDC)
Requirements for zEnterprise Data Compression

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

To support the GDPS® Continuous Availability with zero data loss (GDPS Continuous Availability with zero data loss) solution, Db2 12 introduces subsystem parameters, commands, use of a new data set to support compressed log records, a new REORG TABLESPACE option, enhanced DISPLAY LOG command output, and new IFI qualification field values.

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

- The expansion dictionaries for Db2 tables whose changes are captured

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

Subsystem parameters

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

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

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

CDDS creation and population

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

Commands for starting and stopping the CDDS

Before a CDDS can be recovered, it must be closed and deallocated. The -STOP CDDS command lets you close and deallocate a CDDS without stopping all members of a data sharing group. You can then recover the CDDS, and issue the -START CDDS command to allocate and open the CDDS on all members of the data sharing group. You can issue the -START CDDS and -STOP CDDS commands from the source or proxy data sharing group.

DISPLAY LOG command output changes

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

DSNJU008 (print CDDS) utility

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

IFI qualification changes

> PSPI

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

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

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

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

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

PSPI

Related tasks

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

Related reference

CDDS_MODE in macro DSN6LOGP (Db2 Installation and Migration)

CDDS_PREFIX in macro DSN6LOGP (Db2 Installation and Migration)

-START CDDS (Db2) (Db2 Commands)

-STOP CDDS (Db2) (Db2 Commands)

Qualification fields for READS requests (Db2 Performance)

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

Chapter 6. Query performance in the Db2 12 initial release

The initial release of Db2 12 delivers enhanced query performance.

Important: Db2 12 introduces continuous delivery of new capabilities and enhancements in function levels. Most new capabilities become available only after activation of the Db2 12 function level that introduces them, or when applications run with the corresponding application compatibility level. For more information, see Chapter 12, "Activating Db2 12 function levels," on page 281.

Related concepts

Dynamic plan stability
Db2 12 introduces the capability to achieve access path stability comparable to static SQL statements for repeating cached dynamic SQL statements.

Automated statistics collection
Db2 12 introduces several enhancements that help to automate the collection of statistics.

Related information

Db2 12 for z/OS Optimizer
Db2 12 for z/OS Performance Topics (IBM Redbooks)
IBM Db2 12 for z/OS Technical Overview (IBM Redbooks)

Automated statistics collection

Db2 12 introduces several enhancements that help to automate the collection of statistics.

In Db2 12, you can specify that statistics profiles are updated based on the statistics recommendations that are generated during the optimization process, which were introduced by Db2 11. Profiles are created or updated based on these recommendations when the STATFDBK_PROFILE subsystem parameter values is set to YES.

If you then specify the USE PROFILE option when you collect statistics, the recommended statistics are collected. Db2 12 also introduces support for the USE PROFILE option when you collect statistics inline when you run the following utilities:

- LOAD

- REBUILD INDEX

- REORG TABLESPACE

FL 507 Beginning in function level 507, when statistics are collection with the RUNSTATS TABLESPACE, REORG TABLESPACE, or LOAD utilities with the USEPROFILE option, the utility also deletes any existing statistics that are not included in the profile.

Db2 12 also introduces recommendations from the optimization process to collect statistics that might have become stale, before problems arise. The recommendations are issued for column, multi-column cardinality, frequency, and histogram statistics. The recommendations are issued when STATSTIME value for a statistic particular statistic is older (by a threshold) than the STATSTIME values for all objects that might collect the same statistic. Like other recommendations from the optimization process, they can be automatically applied to statistics profiles.

Also new in Db2 12, if you include the FREQVAL keyword and omit the COUNT keyword, Db2 automatically determines the number of frequently occurring values to collect so that the data distribution is no longer skewed.

Related tasks

Applying statistics recommendations to statistics profiles automatically (Db2 Performance)
Controlling the collection of statistics feedback (Db2 Performance)

Related reference

Related information

Static plan stability enhancements

Db2 12 introduces improvements to the usability of static plan stability features.

Db2 12 introduces the following static plan stability enhancements:

- You can now specify that the FREE PACKAGE command free only the previous or original packages copies by specifying PREVIOUS or ORIGNAL for the PLANMGMTSCOPE option.
- If you specify INACTIVE, PREVIOUS, or ORIGINAL for the PLANMGMTSCOPE option, the FREE PACKAGE command can now complete while applications that use the package are still running, removing the necessity of application outages for maintenance of inactive package copies.
- Improved explain information for APREUSE and APCOMPARE for static plan stability (see columns added to DSN_STATEMNT_TABLE).
- Preservation of LASTUSED for BIND REPLACE of the same version.
- REBIND PACKAGE provides a new option APREUSESOURCE to allow APREUSE to use a PREVIOUS or ORIGINAL copy explicitly.
- Support for statement concentration via a BIND parameter
- Statement concentration for CREATE and ALTER

Related concepts

Db2 12 extends the scope of statement concentration with literals.

Related tasks

Related reference

Related information

Query performance enhancements

Db2 12 introduces performance enhancements for queries that use any of the following: outer joins, UNION ALL, archive transparency, system-period temporal tables.

Db2 12 introduces the following enhancements to query performance:

- Trimming columns from a materialized view or table expression if those columns are not required by the outer query
- Pruning unique LEFT OUTER JOIN views or table expressions if columns are not required in SELECT list
- Pushing join predicates into UNION ALL or outer join query blocks if it is cost effective for the optimizer to do so
- Push ORDER BY and FETCH FIRST into UNION ALL legs to allow each UNION ALL leg to avoid a sort (if possible) and fetch only the required rows for the outer query (for FETCH FIRST)

- Reorder outer join tables to avoid unnecessary materializations
- Reduce situations where workfile usage is required for materialized outer join query blocks or UNION ALL legs

Related concepts

Ways to merge lists of values (Introduction to Db2 for z/OS)

Outer joins (Db2 Application programming and SQL)

Related information

Db2 12 for z/OS Optimizer

User-defined table function performance improvements

The merge capabilities of user-defined table functions are enhanced in Db2 12 to be similar to the capabilities of views.

Db2 12 also introduces indexes on the join or correlation predicates that are passed in as parameters to user-defined table functions to improve performance.

Related concepts

Functions (Db2 SQL)

Related information

Db2 12 for z/OS Optimizer

Improved performance and reliability of index access with list prefetch

Adaptive index is a Db2 12 enhancement to multi-index and single index list prefetch-based plans that introduces logic at execution time to determine the filtering of each index to ensure the optimal execution sequence of indexes, or quicker reversion to table space scan if no filtering index exists.

This enhancement does not require any usage of REOPT bind parameters and therefore avoids any reoptimization overhead at execution time.

Related concepts

Types of indexes (Db2 Administration Guide)

Related information

Db2 12 for z/OS Optimizer

Chapter 7. OLTP performance in the Db2 12 initial release

The initial release of Db2 12 delivers enhanced online transaction processing (OLTP) performance.

Important: Db2 12 introduces continuous delivery of new capabilities and enhancements in function levels. Most new capabilities become available only after activation of the Db2 12 function level that introduces them, or when applications run with the corresponding application compatibility level. For more information, see Chapter 12, "Activating Db2 12 function levels," on page 281.

Related information

Db2 12 for z/OS Performance Topics (IBM Redbooks)

IBM Db2 12 for z/OS Technical Overview (IBM Redbooks)

Improved INSERT performance for unclustered data

Db2 12 introduces *insert algorithm 2* (IAG2), which can improve the performance of INSERT operations for unclustered data.

Using IAG2 can result in an increase in insert throughput, especially with data that is not indexed. IAG2 can also reduce logging and reduce class 2 elapsed time and class 2 CPU time.

IAG2 applies only to universal table spaces (UTS) with the MEMBER CLUSTER attribute. IAG2 does not apply to other table space types.

You can control the insert algorithm used for individual table spaces by specifying the INSERT ALGORITHM option in CREATE TABLESPACE or ALTER TABLESPACE statements. To specify the system-wide setting for default insert algorithm, use the DEFAULT_INSERT_ALGORITHM subsystem parameter.

Recommendation: To determine the group buffer pool size when insert algorithm 2 is used for concurrent INSERT operations from multiple data sharing members, allocate a minimum of 2000 * *n* directory and data entries (where *n* is the number of members).

Related concepts

Insert algorithm 2 (Db2 Performance)

Member affinity clustering (Db2 Data Sharing Planning and Administration)

Related reference

DEFAULT INSERT ALGORITHM field (DEFAULT_INSERT_ALGORITHM subsystem parameter) (Db2 Installation and Migration)

ALTER TABLESPACE (Db2 SQL)

CREATE TABLESPACE (Db2 SQL)

Improved EDM pool management

Db2 12 introduces improved performance for Db2 database systems that have large EDM pools and run multiple, concurrent threads.

These improvements can lead to somewhat higher real storage usage by the database system.

As part of these changes, how EDM pool storage is allocated for certain types of structures has changed.

Previously, the following subsystem parameters determined the amount of storage that was initially allocated in the EDM pool for each type of structure:

- EDMDBDC for the DBD cache
- EDMSTMTC for the dynamic statement cache
- EDM_SKELETON_POOL for the EDM skeleton pool

After the initial allocation, the values could be increased or decreased, but the values could never be less the values that were specified at installation time.

In Db2 12, no storage is initially allocated for these structures. Storage is allocated as it is needed. When the values are reached, and more storage is needed, structures that are not currently in use are removed to keep the storage that is in use below the specified values.

Related reference
EDM DBD CACHE field (EDMDBDC subsystem parameter) (Db2 Installation and Migration)
EDM STATEMENT CACHE field (EDMSTMTC subsystem parameter) (Db2 Installation and Migration)
EDM SKELETON POOL SIZE field (EDM_SKELETON_POOL subsystem parameter) (Db2 Installation and Migration)

In-memory buffer pools

In Db2 12, you can assign objects to in-memory buffer pools. The data pages for the object remain in the buffer pool if they can fit within the size of the buffer pool.

Introductory concepts

Buffer pools (Introduction to Db2 for z/OS)
The role of buffer pools in caching data (Introduction to Db2 for z/OS)

By assigning frequently accessed objects to in-memory buffer pools, you can reduce the cost of internal processing and I/O for those objects.

In-memory buffer pools are created by the PGSTEAL(NONE) option of the ALTER BUFFERPOOL command. Db2 implicitly creates an overflow area as part of the bufferpool. Objects that cannot fit within the in-memory part of the buffer pool are allocated to the overflow area. Page stealing can occur in the overflow area. LRU chains are maintained for the overflow area, but buffer stealing in the overflow area uses the FIFO page-stealing algorithm. When you specify PGSTEAL(NONE), it is best to specify a buffer pool size that is large enough to contain all assigned objects that might be open at the same time. Pages that are allocated to the overflow area cannot be returned to the main area of the bufferpool until the object is closed.

Db2 issues a console message whenever an object is allocated to the overflow area for in-memory buffer pool.

Related tasks
Choosing a page-stealing algorithm (Db2 Performance)
Related reference
-ALTER BUFFERPOOL (Db2) (Db2 Commands)
Related information
DSNB604I (Db2 Messages)

Improvements for the dynamic statement cache

Db2 12 extends the scope of statement concentration with literals.

DB2 10 added an enhancement to dynamic statement caching that allowed a Db2 database manager to exclude the literal constants that are in dynamic SQL statements when it searched for a statement text match in the dynamic statement cache. This enhancement could be specified at the statement level, by including CONCENTRATE STATEMENTS WITH LITERALS in the ATTRIBUTES clause of the PREPARE SQL statement. For ODBC and JDBC applications, the statement could also be specified at the data source level through connection properties.

Db2 12 expands the scope of literal concentration enablement. You can specify statement concentration at the package level, through the CONCENTRATESTMT bind option, and at the routine or advanced trigger level, through the CONCENTRATE STATEMENTS WITH LITERALS option.

Related concepts

Reoptimization for statements with replaced literal values (Db2 Performance)

Related reference

CONCENTRATESTMT bind option (Db2 Commands)

CREATE PROCEDURE (SQL - native) (Db2 SQL)

CREATE FUNCTION (compiled SQL scalar) (Db2 SQL)

CREATE TRIGGER (advanced) (Db2 SQL)

Improved random index access (FTB)

Db2 12 introduces *fast index traversal* (sometimes called "fast traverse blocks" or "FTB"), which can improve overall random index access, especially for indexes used predominantly for read access.

Fast index traversal provides improved performance for indexes with frequent lookups.

In Db2 12, fast index traversal is available in function level 100. However, during coexistence with Db2 11, no storage is allocated for fast index traversal of group buffer pool dependent indexes.

Related concepts

Fast index traversal (Db2 Performance)

Related tasks

Enabling or disabling fast index traversal at the index level (Db2 Performance)

Related reference

INDEX MEMORY CONTROL field (INDEX_MEMORY_CONTROL subsystem parameter) (Db2 Installation and Migration)

Improved lock avoidance checking for data sharing environments

Db2 12 introduces improved lock avoidance checking, which results in better lock avoidance in data sharing environments.

Db2 12 accomplishes this through an increase in the granularity of lock avoidance checking, improving lock avoidance and reducing the traffic to the CF. The result is improved system performance.

For more information about these enhancements, see the following sections in Db2 12 for z/OS Performance Topics (IBM Redbooks):

- "Global commit LRSN enhancement"
- "Global read LSN enhancement"

Related concepts

The Db2 commit process (Diagnosing Db2 problems)

Optimized scheduling of dynamic prefetch

Db2 12 reduces the scheduling of unnecessary prefetch for OLTP and query workloads.

With this feature, Db2 12 addresses the situation that arises when all pages are in memory and dynamic prefetch needlessly schedules a prefetch. The result is a saving of CPU cycles.

Related concepts

Dynamic prefetch (PREFETCH='D') (Db2 Performance)

Reduced log write latency

With z/OS HyperWrite, Db2 12 reduces log write latency, resulting in improved transaction latency.

With this feature, Db2 12 offers a significant opportunity for accelerating transaction execution.

Support for estimating benefits of increasing buffer pool size

Db2 12 provides the buffer pool advisory mode, which simulates the effect on system performance of increasing the size of buffer pools.

With this feature, Db2 Db2 12 enables you to accurately determine how I/O and CPU usage would be affected by increasing buffer pool size.

Related concepts
Buffer pools (Introduction to Db2 for z/OS)

Scalability enhancements

Db2 12 provides several improvements that improve large n-way scaling.

The scalability enhancements introduced in Db2 Db2 12 include:

- Improved efficiency on LPARs with high number of CPs
- Log latch contention reduction
- Bufferpool scaling improvements
- IRLM latch contention reduction
- Increased number of prefetch engines
- Support for > 1 TB buffer pools (sum of VPSIZE and SPSIZE can be up to 16TB)

Related concepts
Db2 data servers and environments (Introduction to Db2 for z/OS)

Increased size of active log data sets

Db2 12 enables support for >4 GB active log data sets.

For Db2 Db2 12, the maximum active log data set is 768 GB.

Adding new active log data set > 4 GB can be done by:

- DSNJU003 stand-alone utility or
- -SET LOG NEWLOG

Related reference
DSNJU003 (change log inventory) (Db2 Utilities)
-SET LOG (Db2) (Db2 Commands)

Instrumentation enhancements

Db2 12 provides several instrumentation enhancements.

These enhancements for Db2 Db2 12 include:

- More granular wait times for IFCIDs 316 (dynamic) and 401 (static)
- Enhanced IFCIDs 53/58 statement level section for PREPARE (similar to INSERT/UPDATE/DELETE)
- Addition of section number in IFCIDs 53/58
- Addition of batch job STEP name in correlation header
- Addition of REFRESH TABLE to counts in DSNDQXST
- Measurement of cache + catalog hit ratio to recognize reduced full prepare due to dynamic plan stability

Related concepts
Instrumentation facilities (IF) (Diagnosing Db2 problems)

Integration with IBM DS8870 Easy Tier Multi-Temperature Management

Db2 12 enhances integration with DS8870 Easy Tier®.

Easy Tier is a "multi-temperature" data placement solution that automatically puts frequently accessed data on SSDs, and less-frequently accessed data on HDDs.

Related concepts
Storage server device planning for Db2 for z/OS (Db2 Performance)

Reduced contention for inserts to PBG table spaces

Db2 12 introduces a new search algorithm for inserts to partition-by-growth (PBG) table spaces to prevent "hot spots" in a single partition.

Starting in Db2 12, a bidirectional search algorithm is used when an insert to a PBG table space fails to find free space in the initial target partition, which is selected based on the clustering index. That is, the partition to be searched next can be either an ascending or descending partition sequence number. Db2 12 randomly select the search order 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.

Note that 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 often uses the last physical partitions, and it can sometimes leave many unused empty partitions unused in between the first and last partitions.

Part 2. Adopting new capabilities in Db2 12 continuous delivery

In Db2 12, 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 V*vv*R*r*M*mmm*, where *vv* is the version, *r* is the release, and *mmm* is the modification level. For example, V12R1M510 identifies function level 510. For a list of all available function levels in Db2 12, see Chapter 2, "Db2 12 function levels," on page 69. Often function level identifiers are abbreviated. For example, " function level 510" refers to V12R1M510.

Tip: GUPI 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 9, "Determining the Db2 code level, catalog level, and function level," on page 271 GUPI

The availability of new function depends on the type of enhancement, the activated function level, and the application compatibility levels of applications. In the initial Db2 12 release, most new capabilities are enabled only after the activation of function level 500 or higher.

Virtual storage enhancements
Virtual storage enhancements become available at the activation of the function level that introduces them or higher. Activation of function level 100 introduces all virtual storage enhancements in the initial Db2 12 release. That is, activation of function level 500 introduces no virtual storage enhancements.

Subsystem parameters
New subsystem parameter settings are in effect only when the function level that introduced them or a higher function level is activated. Many subsystem parameter changes in the initial Db2 12 release take effect in function level 500. For more information about subsystem parameter changes in Db2 12, see "Subsystem parameter changes in Db2 12" on page 200.

Optimization enhancements
Optimization enhancements become available after the activation of the function level that introduces them or higher, and full prepare of the SQL statements. When 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

Activation of function level 100 introduces all optimization enhancements in the initial Db2 12 release. That is, function level 500 introduces no optimization enhancements.

SQL capabilities
New SQL capabilities become available after the activation of the function level that introduces them or higher, for applications that run at the equivalent application compatibility level or higher. New SQL capabilities in the initial Db2 12 release become available in function level 500 for applications that run at the equivalent application compatibility level or higher. You can continue to run SQL statements compatibly with lower function levels, or previous Db2 releases, including Db2 11 and DB2 10. For details, see Application compatibility levels in Db2 12 (Db2 Application programming and SQL)

Procedure

To manage the adoption of new capabilities in Db2 12, 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 12 function levels use the following Db2 catalog levels:

   - V12R1M509
   - V12R1M507
   - V12R1M505
   - V12R1M503
   - V12R1M502
   - V12R1M500

   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 12 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 12
Db2 12 for z/OS takes Db2 to a new level, both extending the core capabilities and empowering the future. Db2 12 extends the core with new enhancements to scalability, reliability, efficiency, security, and availability. Db2 12 also empowers the next wave of applications in the cloud, mobile, and analytics spaces.

Application compatibility levels in Db2 12 (Db2 Application programming and SQL)
Related tasks
Managing application incompatibilities (Db2 Application programming and SQL)

Related reference

Db2 12 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

Exploring IBM Db2 for z/OS Continuous Delivery (IBM Redpaper)
IBM Db2 12 for z/OS Technical Overview (IBM Redbooks)
Video: Db2 for z/OS—Delivering New Capabilities Faster (YouTube: 1:33:25)
PI70406: Add ACTIVATE mode to the Db2 installation CLIST

Chapter 8. Function levels and related levels in Db2 12

Enhancements to Db2 12 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: GUPI 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 9, "Determining the Db2 code level, catalog level, and function level," on page 271 GUPI

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, V12R1M510 identifies function level 510. For a list of all available function levels in Db2 12, see Chapter 2, "Db2 12 function levels," on page 69.

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 12 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 12 function levels use the following Db2 catalog levels:

- V12R1M509
- V12R1M507
- V12R1M505
- V12R1M503
- V12R1M502
- V12R1M500

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 V*vv*R*r*M*mmm*, where *vv* is the version, *r* is the release, and *mmm* is the modification level. For example function level 510 requires catalog level V12R1M509.

An exception is function level 100, which requires catalog level V12R1M500. However, catalog level V12R1M500 is the result when you tailor the catalog for migration to Db2 12, as described in Installation step 16: Tailor the Db2 catalog: DSNTIJTC (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 12 are identified by the same identifier as the corresponding function level. The format is V*vv*R*r*M*mmm*, where *vv* is the version, *r* is the release, and *mmm* is the modification level.

Db2 12 supports the following application compatibility levels in most contexts:

V*vv*R*r*M*mmm*

Compatibility with the behavior of the identified Db2 function level. For example, V12R1M510 specifies compatibility with the highest available Db2 12 function level. The equivalent function level or higher must be activated. For a list of supported Db2 12 function levels, see Chapter 2, "Db2 12 function levels," on page 69.

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. After migration to Db2 12, this value has the same result as specifying V12R1M100.

V10R1

Compatibility with the behavior of DB2 10 new-function mode.

For more information about application compatibility levels, see Application compatibility levels in Db2 12 (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. In some ways function level 100* in Db2 12 is analogous to the CM* and ENFM* migration modes in previous releases. 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. V12R1M500* 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 12 (Db2 Application programming and SQL)

Related reference

Db2 12 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 9. 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

GUPI 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 12:

```
-DB2A DISPLAY GROUP DETAIL
```

Example: Data sharing group with coexisting Db2 12 and Db2 11 members

In the following example, two members are migrated to Db2 12 and ready for the activation of new function in Db2 12. However, the code level of member DB2B indicates that it is not migrated to Db2 12, which means the group is not ready for activation of function level 500. The DB2 LVL value is 111500, which indicates the code level of Db2 11 in new-function mode.

Also, the migration readiness report shows that the Db2 data sharing group is not ready for migration to Db2 13 due to coexistence with Db2 11.

```
-DISPLAY GROUP DETAIL
DSN7100I  -DB2A DSN7GCMD
*** BEGIN DISPLAY OF GROUP(DSNCAT  ) CATALOG LEVEL(V12R1M500)
                CURRENT FUNCTION LEVEL(V12R1M100)
                HIGHEST ACTIVATED FUNCTION LEVEL(V12R1M100)
                HIGHEST POSSIBLE FUNCTION LEVEL(V12R1M100)
                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    121500 MVSA      DJ2A   DB2AIRLM
DB2B        2 DB2B -DB2B      ACTIVE    111500 MVSB      DJ2B   DB2BIRLM
DB2C        3 DB2C -DB2C      ACTIVE    121500 MVSC      DJ2C   DB2CIRLM
 -------------------------------------------------------------------

MIGRATION READINESS REPORT
--------------------------
DB2         CODE        SPE      MIGRATION
MEMBER      LEVEL       APAR     ELIGIBLE
--------    ---------   -------  ---
DB2A        V12R1M500   .......  NO
DB2B        V11R1M500   .......  NO
DB2B        V12R1M500   .......  NO
 -------------------------------------------------------------------
MIGRATION READINESS STATUS: IS NOT READY FOR DB2 13
REASON: COEXISTENCE PRECLUDES MIGRATION TO THE NEXT DB2 RELEASE
 -------------------------------------------------------------------
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: Data sharing group with all active members migrated to Db2 12 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 12 and ready for the activation of function level 500.

Also, the migration readiness report shows that the Db2 data sharing group is not ready for migration to Db2 13 due to the highest activated function level and not all members have the fallback SPE APAR.

```
DSN7100I  -DB2B DSN7GCMD
*** BEGIN DISPLAY OF GROUP(DSNCAT  ) CATALOG LEVEL(V12R1M500)
                  CURRENT FUNCTION LEVEL(V12R1M100)
                  HIGHEST ACTIVATED FUNCTION LEVEL(V12R1M100)
                  HIGHEST POSSIBLE FUNCTION LEVEL(V12R1M500)
                  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   121500 MVSA     DJ2A   DB2AIRLM
DB2B       2 DB2B -DB2B     ACTIVE   121510 MVSB     DJ2B   DB2BIRLM
DB2C       3 DB2C -DB2C     ACTIVE   121500 MVSC     DJ2C   DB2CIRLM
-----------------------------------------------------------------------
MIGRATION READINESS REPORT
-------------------------
DB2        CODE       SPE      MIGRATION
MEMBER     LEVEL      APAR     ELIGIBLE
-------- --------- ------- ---
DB2A      V12R1M510  .......  NO
DB2B      V12R1M510  PH37108  NO
DB2C      V12R1M510  .......  NO
-----------------------------------------------------------------------
MIGRATION READINESS STATUS: GROUP IS NOT READY FOR DB2 NEXT
REASON: HIGHEST ACTIVATED FUNCTION LEVEL NOT V12R1M510
-----------------------------------------------------------------------
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 12 after the activation of function level 500

The following DISPLAY GROUP output illustrates a data sharing group with function level 500 activated.

Also, the migration readiness report shows that the Db2 data sharing group is not ready for migration to Db2 13 due to the highest activated function level and the missing SPE APAR.

```
DSN7100I  -DB2A DSN7GCMD
*** BEGIN DISPLAY OF GROUP(DSNCAT  ) CATALOG LEVEL(V12R1M500)
                  CURRENT FUNCTION LEVEL(V12R1M500)
                  HIGHEST ACTIVATED FUNCTION LEVEL(V12R1M500)
                  HIGHEST POSSIBLE FUNCTION LEVEL(V12R1M500)
                  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   121500 MVSA     DJ2A   DB2AIRLM
```

```
DB2B        2 DB2B -DB2B     ACTIVE   121500 MVSB      DJ2B    DB2BIRLM
DB2C        3 DB2C -DB2C     ACTIVE   121500 MVSC      DJ2C    DB2CIRLM
----------------------------------------------------------------------
MIGRATION READINESS REPORT
--------------------------
DB2         CODE       SPE     MIGRATION
MEMBER      LEVEL      APAR    ELIGIBLE
--------    ---------  ------- ---
DB2A        V12R1M500  ....... NO
DB2B        V12R1M500  ....... NO
DB2B        V12R1M500  ....... NO
----------------------------------------------------------------------
MIGRATION READINESS STATUS:  GROUP IS NOT READY FOR DB2 13
REASON: HIGHEST ACTIVATED FUNCTION LEVEL NOT V12R1M510
----------------------------------------------------------------------
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: Db2 12 data sharing group ready for migration to Db2 13

The example single-member Db2 12 data sharing group is ready for migration to Db2 13.

```
DSN7100I  -DB2A DSN7GCMD
*** BEGIN DISPLAY OF GROUP(DSNCAT  ) CATALOG LEVEL(V12R1M509)
                  CURRENT FUNCTION LEVEL(V12R1M510)
                  HIGHEST ACTIVATED FUNCTION LEVEL(V12R1M510)
                  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 PRLM2171
DB2A        1 DB2A -DB2A     ACTIVE   121510 UTEC5     PR21    PRLM21
----------------------------------------------------------------------
MIGRATION READINESS REPORT
--------------------------
DB2         CODE       SPE     MIGRATION
MEMBER      LEVEL      APAR    ELIGIBLE
--------    ---------  ------- ---
DB2A        V12R1M510  PH37108 YES
----------------------------------------------------------------------
MIGRATION READINESS STATUS: GROUP IS READY FOR MIGRATION TO DB2 13
----------------------------------------------------------------------
DISPLAY SUBGROUP ATTACH INFORMATION FOR GROUP ATTACH DSNG
----------------------------------------------------------------------
----------------------------------------------------------------------
SCA   STRUCTURE SIZE:    12288 KB, STATUS= AC,   SCA IN USE:     5 %
LOCK1 STRUCTURE SIZE:    12288 KB
NUMBER  LOCK ENTRIES:     1048576
NUMBER  LIST ENTRIES:       23151, LIST ENTRIES  IN USE:          0
SPT01 INLINE LENGTH:        32138
*** END DISPLAY OF GROUP(DSNCAT  )
-DB2A DSN7GCMD 'DISPLAY GROUP ' NORMAL DSN9022I  COMPLETION
```

Example: Db2 12 data sharing group not ready for migration to Db2 13

The example Db2 12 data sharing group not ready for migration to Db2 13 because of the coexistence with Db2 11.

```
DSN7100I  -DB2A DSN7GCMD
*** BEGIN DISPLAY OF GROUP(DSNCAT  ) CATALOG LEVEL(V12R1M500)
                  CURRENT FUNCTION LEVEL(V12R1M100)
                  HIGHEST ACTIVATED FUNCTION LEVEL(V12R1M100)
                  HIGHEST POSSIBLE FUNCTION LEVEL(V12R1M100)
                  PROTOCOL LEVEL(2)
                  GROUP ATTACH NAME(DSNG)
----------------------------------------------------------------------
DB2         SUB                     DB2     SYSTEM    IRLM
MEMBER   ID SYS CMDPREF   STATUS    LVL     NAME      SUBSYS PRLM2171
DB2A        1 DB2A -DB2A     ACTIVE   121510 UTEC5     PR21    PRLM21
DB2B        2 DB2B -DB2B     ACTIVE   111500 UTEC5     QR21    QRLM21
----------------------------------------------------------------------
```

```
-------------------------------------------------------------------
DISPLAY SUBGROUP ATTACH INFORMATION FOR GROUP ATTACH DSNG
-------------------------------------------------------------------
MIGRATION READINESS REPORT
--------------------------
DB2       CODE       SPE      MIGRATION
MEMBER    LEVEL      APAR     ELIGIBLE
--------  ---------  -------  ---
DB2A      V12R1M510  PH37108  YES
DB2B      V11R1M500  .......  NO
-------------------------------------------------------------------
MIGRATION READINESS STATUS: GROUP IS NOT READY FOR DB2 13
REASON: COEXISTENCE PRECLUDES MIGRATION TO THE NEXT DB2 RELEASE
-------------------------------------------------------------------
SCA   STRUCTURE SIZE:     12288 KB, STATUS= AC,   SCA IN USE:     5 %
LOCK1 STRUCTURE SIZE:     12288 KB
NUMBER  LOCK ENTRIES:     1048576
NUMBER  LIST ENTRIES:       23151, LIST ENTRIES  IN USE:         5
SPT01 INLINE LENGTH:       32138
*** END DISPLAY OF GROUP(DSNCAT  )
DSN9022I  -DB2A DSN7GCMD 'DISPLAY GROUP ' NORMAL COMPLETION
```

Example: standalone Db2 12 subsystem not ready for migration to Db2 13

The example Db2 subsystem has a sufficient code level and the fallback SPE APAR for migration to Db2 13. However, it is not ready for migration to Db2 13 because of its highest activated function level.

```
DSN7100I  -DB2A DSN7GCMD
*** 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
-------------------------------------------------------------------
-------------------------------------------------------------------
DISPLAY SUBGROUP ATTACH INFORMATION FOR GROUP ATTACH DSNG
-------------------------------------------------------------------
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
-------------------------------------------------------------------
SPT01 INLINE LENGTH:       32138
*** END DISPLAY OF GROUP(........)
DSN9022I  -DB2A DSN7GCMD 'DISPLAY GROUP ' NORMAL COMPLETION
```

FL 502 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 DSN7GCMD
*** BEGIN DISPLAY OF GROUP(DSNCAT  ) CATALOG LEVEL(V12R1M502)
              CURRENT FUNCTION LEVEL(V12R1M502)
              HIGHEST ACTIVATED FUNCTION LEVEL(V12R1M502)
              HIGHEST POSSIBLE FUNCTION LEVEL(V12R1M502)
              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 12, "Activating Db2 12 function levels," on page 281.

Related concepts

Function levels and related levels in Db2 12
Enhancements to Db2 12 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 10. 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 9, "Determining the Db2 code level, catalog level, and function level," on page 271.

Procedure

To test activation of a Db2 function level, complete the following steps:

1. GUPI Issue an ACTIVATE command with the TEST option and specify the target Db2 function level to test.

   ```
    -ACTIVATE FUNCTION LEVEL (V12R1M500) 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 (V12R1M500)
                GROUP ELIGIBLE FOR FUNCTION LEVEL (V12R1M500)
                CATALOG LEVEL(V12R1M500)
                CURRENT FUNCTION LEVEL(V12R1M100)
                HIGHEST ACTIVATED FUNCTION LEVEL (V12R1M100)
                HIGHEST POSSIBLE FUNCTION LEVEL(V12R1M500)
   -----------------------------------------------------------------
   DB2          CURRENT    CAPABLE FUNCTION LEVELS
   MEMBER   ID  CODE-LEVEL LOWEST     HIGHEST     STATUS
   -------- --- ---------- ---------- ----------  -------------------
   DB2A       1 V12R1M500  V12R1M100  V12R1M500   ELIGIBLE
   DB2B       2 V12R1M500  V12R1M100  V12R1M500   ELIGIBLE
   DB2C       3 V12R1M500  V12R1M100  V12R1M500   ELIGIBLE
   -----------------------------------------------------------------
   DSN9022I  -DB2A DSNZACMD '-ACTIVATE FUNC' NORMAL COMPLETION
   ```

   GUPI

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 12, "Activating Db2 12 function levels," on page 281.

Related concepts

Function levels and related levels in Db2 12

Enhancements to Db2 12 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 11. 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 12. Activating Db2 12 function levels

You control the activation and adoption of new features in Db2 12 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

Restriction: Before you can run CATMAINT to tailor the Db2 catalog for function level 502 or higher, you must first activate function level 500 or 501. That is, the DISPLAY GROUP command output must indicate `HIGHEST ACTIVATED FUNCTION LEVEL (V12R1M500)` or higher. This restriction prevents tailoring the Db2 catalog for function levels higher than 500 while fallback to Db2 11 remains possible. However, later activating a lower function level such as function level 100* does not restrict the CATMAINT operation.

Ensure that no incompatible applications will interfere with the catalog update. For details, see Chapter 11, "Identifying applications that are incompatible with catalog updates," on page 279.

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 2, "Db2 12 function levels," on page 69
-
- "Incompatible changes summary for function levels 501 and higher" on page 119
- Chapter 3, "Changes to plan for in Db2 12," on page 119

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 13, "Activating Db2 function levels by using z/OSMF," on page 285.

Procedure

To activate capabilities and enhancements that are introduced by Db2 function levels, complete the following steps:

1. If Db2 12 is at function level 100, follow the steps in Activating Db2 12 new function at migration (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 9, "Determining the Db2 code level, catalog level, and function level," on page 271.
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 Db2 installation CLIST.

   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 12, or most recently activated a Db2 12 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 V12R1M5*nn* in the TARGET FUNCTION LEVEL field, where 5*nn* is the function level that you want to activate. The format is V*vv*R*r*M*mmm*, 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 V12R1M5*nn* 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.

   Restriction: Before you can run CATMAINT to tailor the Db2 catalog for function level 502 or higher, you must first activate function level 500 or 501. That is, the DISPLAY GROUP command output must indicate HIGHEST ACTIVATED FUNCTION LEVEL (V12R1M500) or higher. This restriction prevents tailoring the Db2 catalog for function levels higher than 500 while fallback to Db2 11 remains possible. However, later activating a lower function level such as function level 100* does not restrict the CATMAINT operation.

   When the catalog level change completes, Db2 records the change in the SYSIBM.SYSLEVELUDPATES catalog table and issues message DSNG014I 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

8. Run the tailored DSNTIJAF job, or issue the following ACTIVATE command, with V12R1M5*nn* changed to function level that you want to activate.

```
-ACTIVATE FUNCTION LEVEL (V12R1M5nn)
```

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.

GUPI For example, the following message indicates successful activation:

```
DSNU757I  -DB2A DSNUGCCA
*** BEGIN ACTIVATE FUNCTION LEVEL (V12R1M502)
```

```
                  FUNCTION LEVEL (V12R1M502) SUCCESSFULLY ACTIVATED
                  CATALOG LEVEL(V12R1M502)
                  CURRENT FUNCTION LEVEL(V12R1M500)
                  HIGHEST ACTIVATED FUNCTION LEVEL (V12R1M500)
                  HIGHEST POSSIBLE FUNCTION LEVEL(V12R1M502)
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 DSNG014I to the console.

◀ **GUPI**

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 15, "Controlling the Db2 application compatibility level," on page 289.

   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 15, "Controlling the Db2 application compatibility level," on page 289.

- If you encounter regressions or other problems when you activate a Db2 12 function level, minimize the impact to your applications while you resolve the problems by following the general approaches described in Chapter 16, "Responding to problems after function level activation," on page 291.

- 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 12, see "Catalog changes in Db2 12" on page 179.

Related concepts

Function levels and related levels in Db2 12
Enhancements to Db2 12 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 12 (Db2 Application programming and SQL)

Related reference

Db2 12 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 13. 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 12 new function at migration (Db2 Installation and Migration).

Ensure that no incompatible applications will interfere with the catalog update. For details, see Chapter 11, "Identifying applications that are incompatible with catalog updates," on page 279.

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 2, "Db2 12 function levels," on page 69
-
- "Incompatible changes summary for function levels 501 and higher" on page 119
- Chapter 3, "Changes to plan for in Db2 12," on page 119

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 9, "Determining the Db2 code level, catalog level, and function level," on page 271.

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 12 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 12 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 14. 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 (DSNZP*xxx*), 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 15. 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 12 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 2, "Adopting new capabilities in Db2 12 continuous delivery," on page 261.

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 12 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 12), continue to run your applications at the same application compatibility level until you are satisfied that your Db2 12 environment is stable at the new function level.
2. Rebind the packages for SPUFI and the productivity-aid sample programs (such as DSNTEP2, DSNTEP4, 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 12 (Db2 Application programming and SQL)
Function levels and related levels in Db2 12
Enhancements to Db2 12 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 12 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 16. Responding to problems after function level activation

If you encounter regression or other problems after the activation of a new Db2 12 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. In some ways function level 100* in Db2 12 is analogous to the CM* and ENFM* migration modes in previous releases. 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. V12R1M500* 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 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 12 migration and function level activation processes from application updates to adopt new features and resolve incompatible changes.

Chapter 17. 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 501 Throughout the Db2 12 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 12 continuous delivery
In Db2 12, function levels and application compatibility levels control the adoption of most new capabilities by Db2 subsystems and Db2 applications.

Related reference

Db2 12 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 12 for z/OS and related products in IBM Documentation.

For all online product documentation for Db2 12 for z/OS, see IBM Documentation (https://www.ibm.com/docs/en/db2-for-zos/12).

For other PDF manuals, see PDF format manuals for Db2 12 for z/OS (https://www.ibm.com/docs/en/db2-for-zos/12?topic=zos-pdf-format-manuals-db2-12).

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 12 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 12 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 12 for z/OS.

General-use Programming Interface and Associated Guidance Information is identified where it occurs by the following markings:

<kbd>GUPI</kbd> General-use Programming Interface and Associated Guidance Information... <kbd>GUPI</kbd>

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:

<kbd>PSPI</kbd> Product-sensitive Programming Interface and Associated Guidance Information... <kbd>PSPI</kbd>

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

IFCID (instrumentation facility component identifier)
changed 193
new 193
IFCID 0366 167
IFCID 0376 167
in-memory buffer pools 256
incompatibilities of releases
applications and SQL 123
commands 139
security 140
unsupported functions 216
utilities 134
incompatible applications
identifying 170, 279
index compression
altering 237
insert algorithm 2 255
insert partition 241
installation
installation SYSOPR 236
SYSADM 236

J

JSON 228

K

KEEPDYNAMIC bind option
ROLLBACK 126

L

links
non-IBM Web sites
298
LRSN
extended 209

M

macro DSN6SPRM
AUTH_COMPATIBILITY 242
MERGE statement
enhancements 229
migration
considerations 119
improved 235
installation SYSOPR 236
SYSADM 236

O

ODBC
enhancements 232
ownership
transferring
TRANSFER OWNERSHIP statement 239

P

packages

packages *(continued)*
automatic rebinds 126
partition-by-range table spaces 241
pending definition changes
column alterations 237
recovery prior to materialization 237
percentile functions 231
performance enhancements
buffer pools 256
plans
automatic rebinds 126
point-in-time recovery
prior to materialization of pending definition changes
237
POWER built-in function 128
product-sensitive programming information, described 298
programming interface information, described 298
PSPI symbols 298
pureXML
multiple XML updates 228

R

RBA
extended 209
real-time statistics improvements 246, 255
rebinds
automatic 126
release incompatibilities
applications and SQL 123
commands 139
security 140
unsupported functions 216
utilities 134
removed functions 216
REPAIR
CATALOG 239
resource limit facility
static SQL statements 233
routines
array parameters 227, 228

S

security release incompatibilities 140
SQL PL
in trigger 227
SQL release incompatibilities 123
SQL statements
changes 152
TRANSFER OWNERSHIP 239
SQL variables
arrays 227, 228
static SQL statements
resource limit facility 233
stored procedures
array parameters 227, 228
Support for maintaining session data on the target server
233

T

temporal logical transactions 231

temporal referential constraint 230
trace field QW0366FN 167
trace field QW0376FN 167
TRANSFER OWNERSHIP statement
    database objects 239
    system objects 239
triggers
    SQL PL 227

U

unsupported functions 216
utilities
    changes 172
    release incompatibilities 134

V

V12R1M507 77
V12R1M508 75
V12R1M509 71
V12R1M510 69

X

XMLMODIFY
    multiple XML updates 228

IBM ®

Product Number: 5650-DB2
                  5770-AF3