

Exploring IBM Db2 for z/OS Continuous Delivery

Chris Crone



Information Management



International Technical Support Organization

Exploring IBM Db2 for z/OS Continuous Delivery

March 2018

Note: Before using this information and the product it supports, read the information in “Notices” on page v.

First Edition (March 2018)

This edition applies to Version 12, Release 1, Modification 00 of Db2 12 for z/OS (5650-DB2).

This document was created or updated on March 21, 2018.

© Copyright International Business Machines Corporation 2018. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	v
Trademarks	vi
Preface	vii
Authors	vii
Now you can become a published author, too!	vii
Comments welcome	viii
Stay connected to IBM Redbooks	viii
Chapter 1. Continuous delivery for Db2 12 for z/OS	1
1.1 Why continuous delivery	2
1.1.1 Proposals for implementing continuous delivery	3
1.1.2 The strategy	3
1.2 Migration from Db2 11 to Db2 12	6
1.2.1 Function levels	7
1.2.2 Relationship between code levels and function levels	11
1.2.3 Relationship between a function level and APPLCOMPAT	11
1.2.4 Catalog levels	12
1.2.5 Migrating Db2 11 data sharing groups to Db2 12	13
1.3 New function activation beyond base Db2 12 capability	14
1.3.1 Activate new function without catalog changes	14
1.3.2 Activate new function with catalog changes	15
1.3.3 Activate new function with catalog change while skipping function levels	16
1.3.4 Function activation of previous level (* mode)	17
1.3.5 SYSLIB.SYSLEVELUPDATES table	19
1.3.6 Additional table changes	20
1.4 DISPLAY GROUP command	21
1.4.1 DISPLAY GROUP (coexistence): From Db2 12 member	22
1.4.2 DISPLAY GROUP (new function not available)	23
1.4.3 DISPLAY GROUP (New Function Active)	24
1.4.4 DISPLAY GROUP (after CATMAINT)	24
1.5 Function level adoption preferred practices	25
1.6 APPLCOMPAT adoption preferred practices	25
Chapter 2. Changes in Db2 12 affecting applications and more	27
2.1 Application considerations	28
2.1.1 The behavior of SQL, DDL, and DCL in Db2 12	29
2.1.2 Dynamic SQL, DDL, and DCL	30
2.2 New built-in global variables	31
2.2.1 PRODUCTID_EXT	31
2.2.2 CATALOG_LEVEL	31
2.2.3 DEFAULT_SQLLEVEL	32
2.3 The IBM Data Server Driver	32
2.3.1 The ClientApplCompat keyword	32
2.3.2 What the clients see after Connect	36
2.3.3 SQLCA and GET DIAGNOSTIC impacts	36
2.3.4 DDF commands and messages	37
Chapter 3. Db2 12 documentation: Information in a new way	39

Chapter 4. Final thoughts on continuous delivery.	47
Related publications	49
IBM Redbooks	49
Online resources	49
Help from IBM	49

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 grant 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

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.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

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.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.


Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, 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>

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

CICS®
Db2®
DB2®
IBM®

IBM z®
IMS™
Redbooks®
Redpaper™

Redbooks (logo) ®
z/OS®

The following terms are trademarks of other companies:

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Preface

This IBM® Redpaper™ publication provides key information about continuous delivery in IBM Db2® 12 for z/OS®. It discusses how continuous delivery works and the changes that have been made in Db2 12 to support continuous delivery, such as adding a new catalog table and changing existing catalog tables. Also the paper covers the effects on applications and how to take advantage of new function provided using the continuous delivery model.

Authors

This paper was produced with the International Technical Support Organization, Raleigh Center.

Chris Crone is an IBM Distinguished Engineer and the team lead for the Relational Database Service (RDS) Execution Engine area in IBM Db2 for z/OS. Chris has worked on Db2 for z/OS for over 28 years. He has contributed to many initiatives in Db2, such as stored procedures, user-defined functions, and multiple-row insert and fetch. Chris works frequently with customers and independent software vendors (ISVs) on building and porting applications to Db2.

Thanks to the following people for their contributions to this project:

Martin Keen, LindaMay Patterson
International Technical Support Organization, Raleigh Center

John Campbell
IBM Distinguished Engineer

Paul McWilliams
Advisory Technical Writer, Db2 for z/OS and Db2 Tools

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our papers to be as helpful as possible. Send us your comments about this paper or other IBM Redbooks® publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

ibm.com/redbooks

- Send your comments in an email to:

redbooks@us.ibm.com

- Mail your comments to:

IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on Facebook:

<http://www.facebook.com/IBMRedbooks>

- Follow us on Twitter:

<http://twitter.com/ibmredbooks>

- Look for us on LinkedIn:

<http://www.linkedin.com/groups?home=&gid=2130806>

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>

- Stay current on recent Redbooks publications with RSS Feeds:

<http://www.redbooks.ibm.com/rss.html>



Continuous delivery for Db2 12 for z/OS

Db2 12 for z/OS delivers the infrastructure to support continuous delivery. With continuous delivery, Db2 ships new function, preventive service, and prescriptive service in the same service.

This chapter provides an overview of the continuous delivery function and includes the following topics:

- ▶ Why continuous delivery
- ▶ Migration from Db2 11 to Db2 12
- ▶ New function activation beyond base Db2 12 capability
- ▶ DISPLAY GROUP command
- ▶ Function level adoption preferred practices
- ▶ APPLCOMPAT adoption preferred practices

1.1 Why continuous delivery

IBM Db2 12 for z/OS has many new innovations that increase Db2 availability, reliability, scalability, and security for business critical information. It offers performance and functional improvements for both transactional and analytical workloads and makes installation and migration simpler and faster. It also allows you to develop applications for the cloud and mobile devices by providing self-provisioning, multitenancy, and self-managing capabilities in an agile development environment.

With Db2 12 and its support for the continuous delivery function, preventive service, and prescriptive service ship in the same stream. IBM has used this approach before; in fact, Db2 always shipped some new function in the service stream. Two examples of this approach are:

- ▶ Db2 10 introduced support for IBM Db2 Analytics Accelerator support in the service stream.
- ▶ Db2 11 provided JavaScript Object Notation (JSON) support and native RESTful service APIs in the service stream.

What is new is that IBM announced its intention to deliver significant, mainline enhancements in the service stream. In the past, Db2 development generally tried to limit enhancements in the service stream to periphery features when possible. This approach balanced the need to bring new function to market and maintain product stability.

To stay competitive, enterprises are frequently demanding that applications support new function and capabilities in days, not months or years. This requirement makes access to the latest Db2 functions and features vital. The question of how to best address this requirement was the subject of much discussion between Db2 development and customers.

Prior to Db2 12, the status quo for releases was as follows:

- ▶ The Db2 development team produced new releases approximately every 3 years, but customer adoption of these new releases was longer than 3 years. This adoption lag time caused customers to be a whole release behind after a few releases. Because customers were not in sync on which releases they had adopted, the Db2 team was asked to support skip release migration for literally every release. Although the Db2 team recognized the need for skip release migration, it was not necessarily in the best interest of Db2 or customers for customers to remain two releases back level. Therefore, skip release migrations were supported roughly only once every 10 years.
- ▶ In every version the Db2 development team retrofitted a limited number of new features in the service stream, which were primarily urgent and generally low risk features. This approach forced many customers to wait for needed features.
- ▶ Customers saw deployment of new releases as disruptive, making them reluctant to move to a new release, which inhibited their ability to take advantage of new features.
- ▶ Many customers wanted new features delivered much faster.
- ▶ Industry and customer trends are moving away from monolithic code delivery toward a *continuous delivery model*.

1.1.1 Proposals for implementing continuous delivery

The Db2 team reviewed the following proposals for continuous delivery with customers:

- ▶ Support a faster release cycle with new Db2 releases every 18 months.
 - Customers were not able to take on such a fast migration due to the cost of any version or release migration.
 - The migration effort would be just as large as moving to a new Db2 version.
 - A new version or release is a new function modification identifier (FMID), ordering a new FMID requires justification, a project plan, and funding in most customer shops.
 - Even with the promise of skip release migration and no catalog changes in the vNext release, the concept was still unacceptable.
- ▶ Support continuous release cycles with each release (such as, Db2 12.1, 12.2, and 12.3) where the customer orders Db2 12 and gets the current point release that is available at that time. For example, if a customer ordered Db2 12 3 years after general availability (GA), the customer gets Db2 12.x, where x is the release level that was current at that time.

Customer also did not accept this approach because it generated considerable migration and coexistence challenges and a high cost of maintenance (forward fits) for the Db2 development organization.

- ▶ Support a combination of service only (sometimes referred to as a *maintenance only* or *stable release*) and new function release streams. Although many other products are pursuing this path, in a Db2 data sharing group, all members must be at the same level to use a new function. So, a V12.1.1 stable release version of Db2 cannot coexist with a V12.4.1 new function release Db2, and if they did, capability is limited to the V12.1.1 release. Integration of a new function release with a stable function release is just a new migration, which was also not acceptable.

Ultimately continuous delivery of a single stream was determined to be the best way to deliver new function. The feedback from customers was clear. Integration of new function in the service stream is the best strategy, if adoption of function can be controlled by the customer and if applying maintenance will not require activation of new function.

1.1.2 The strategy

The starting point for continuous delivery was GA of Db2 12. Starting with Db2 12, customers will see a significantly higher number of new function APARs that deliver new capability in the service stream. There is a single maintenance stream for Db2 12, which includes corrective service, preventative service, and PTFs that are used to enable new function.

This approach allows customers to adopt new function in small, consumable units at a pace they are comfortable with. A new release can be a big effort for our customers. Db2 introduces function levels and extends *application compatibility* (referred to as APPLCOMPAT hereafter) to support function levels.

APPLCOMPAT can be used to test and deploy applications at a specific function level and to stabilize the functional environment for that application, while still allowing other applications to take advantage of recently introduced features that are provided by higher-level function levels. This approach enables customers to *establish islands of stability for applications*.

With continuous delivery, customers can run Db2 code that is capable of doing things that have not been adopted yet. By running the “maintenance” that supports a function level, before adopting that function level (using the **-ACTIVATE** command), customers can ensure

that the code level is solid before adopting the function level that the code supports. *Activating a function level does not do anything other than allow you to use the new function.* Applications remain stable because their behavior is controlled by APPLCOMPAT, not by a function level.

The Db2 team expects that point releases will be a rare exception. However, there are situations when a new point release or a new version might be necessary. For example, occasionally Db2 adopts a new compiler, which can cause the recompilation of large portions of the product code. Recompiling the product is not something that makes sense to do in the maintenance stream (one or more APARs), because that can create a maintenance level set that makes it difficult to apply a needed APAR in a situation where a fix is needed outside of a normal maintenance cycle. Another example is the need to prerequisite a new IBM z® Systems architecture level or z/OS version. Such prerequisites are usually associated with architectural changes that affect significant portions of the product and, as such, are probably better introduced at a version or release boundary.

As part of continuous delivery, Db2 development is committed to maintaining production level quality in the service stream. Db2 development understands the need for system stability and availability. The Db2 team is going to control the input to development in terms of the size and risk of the items to be delivered under continuous delivery. Any new function produced will not block the teams' ability to deliver service. There is an increased focus on functional and performance regression testing in support of the maintenance stream. New function is delivered when the quality is correct and the item is completely tested, not based on a delivery date. Items not included in Db2 12, new initiatives, and request for enhancements (RFEs) are candidates for future delivery. As always, an item can be in the backlog, without a commitment to deliver that item; however, that simply means the item has enough merit to be considered.

The Db2 internal resource lock manager (IRLM) is both a separate subsystem and an integral component of Db2. IRLM works with Db2 to ensure data integrity through serialization. IRLM has been using a continuous delivery strategy for many years, so there are no changes to IRLM.

Continuous delivery in Db2 12 solves customer issues concerning moving to a new release in the following ways:

- Reduce the time to adopt new functionality

Customers can run with the Db2 code that contains the new function as part of their normal maintenance process. By doing this, the customer is assured that the code that supports the function level does not regress existing applications. After a function level is activated, only those applications that need the new function are exposed to that function via APPLCOMPAT. Customers are more likely to have the function that they need already installed on their systems when they need it for an application. All they need to do is activate it.

- Increases the time between versions

This approach drives the delivery of more IT organization value because less time and effort is spent on migrating to new versions of Db2.

Strategy for SMP/E

Companies handle management of SMP/E for z/OS environments for the Db2 run time differently, so one size does not fit all. In general, they *Receive* and *Apply* maintenance, but do not *Accept* the maintenance until they are ready to start the next maintenance cycle. This approach allows the customer to reject any maintenance that might be causing a problem.

At the start of the next maintenance cycle, the customer *Accepts* the previous maintenance and starts over with a new *Receive* and *Apply* cycle. Given this paradigm, it is reasonable to assume that most customers will not activate a function level until they have done an *Accept* on that maintenance. For example, if a customer does two maintenance cycles a year, this equates to a lag of about 6 months between the time that maintenance is applied and the time that a customer considers activating that function level. Of course, function can be made available in development environments in a more aggressive manner, if specific capabilities are needed for a development project.

Note: There are no changes to consolidated service test (CST) and recommended service upgrades (RSUs) done on IBM Db2 maintenance streams. These changes are done after Db2 ships the maintenance release. Both CST and RSU continues as before.

Design Thinking

The Db2 development team is complementing their continuous delivery model by adopting the Design Thinking methodology. A key component to Design Thinking is sponsor users. For example, a group of sponsor users who want a new function are involved early in the development process to make sure development gets the design correct and that this new function is readily consumable. Development wants to deliver a function that is easy to adopt and that customers truly believe in. If there are no sponsor users for a particular function, there will be no development of that function.

Traditionally, development has worked with a small number of customers in the early support program (ESP). In the ESP, customers were involved in validation of what was built. With Design Thinking, the sponsor users are involved when the items are being designed to ensure that the item is designed correctly and is readily consumable. Customer input in the future is going to be much more targeted than what it was in the ESPs. The Db2 development team has a stronger focus on getting the design correct up front.

ISV consideration

Independent software vendors (ISVs), whether they are tool vendors or application vendors, are an important part of the Db2 and IBM Z ecosystem. The Db2 development team realizes the dependency that customers have on ISV products as part of their Db2 environment. Because of these dependencies, the Db2 development team runs a vendor program that provides detailed information about changes and future plans to all vendors on an equal basis. All vendors, including IBM program products, receive the same information about upcoming changes at the same time. Typically, this occurs early in the development process, regardless of whether the change is shipped as part of a new version or as part of an enhancement in the field via an APAR.

For typical version upgrades, the vendor program includes both early and regular communication updates as well as early access to the code. ISV code drops of the new release usually start about a year prior to the ESP start.

The Db2 development team also maintains a list of ISV contacts and a list of specific APIs, modules, or control structures that ISVs are sensitive to. ISVs receive automated updates with notification of APARs affecting interfaces an ISV indicates they are sensitive to. Regular sharing of documentation for design changes delivered in a maintenance stream is done prior to shipping those changes.

ISV continuous delivery support summary

The overriding aim of the Db2 development team is to ensure the stability of customer environments. To help achieve this goal the Db2 development team commits to increased support for ISVs in the following ways:

- ▶ Improved, earlier communication of changes
- ▶ Early access to new maintenance
- ▶ More consistent and timely update of serviceability information
- ▶ Periodic longer view updates
- ▶ Ongoing use of improved communication channels between ISVs and Db2 development

In general, information about a new function in a given function level is made available when that function level ships. However, ISVs have insight into the backlog, and the Db2 development team shares information with them on what is coming in advance.

Continuous delivery places greater responsibility upon ISVs to invest in ensuring stability and preventing regression involving Db2 maintenance. The Db2 development team has had discussions with the vendor community in addition to one-on-one discussions with vendors to ensure that everyone is aware of the changes continuous delivery brings to customers.

The goal of all of this effort is to not break ISV code when an APAR ships. Both the Db2 development team and the ISV community understand that there might be cases where ISVs need to ship enhancements to their code before new capability is activated. However, the goal is for ISVs to depend on function-level activation, not application of APARs. Customers might need to understand ISV dependencies before activation of the capability. This process is similar to situations in the past where ISVs shipped version toleration code, and then later shipped version or feature exploitation code.

Discussions with various ISVs suggested they pick a level (for example, V12R1M500) and certify their software to that function level (for example their version X.y). When an ISV needs a new Db2 function, they change to a new version (such as, version X.z) and test and certify for that function level or application compatibility level (for example V12R1M505). The vendor ships code with the application compatibility level that is needed by that software in the BIND statement of the code. The application compatibility level ensures that the software runs, as tested in their environment, regardless of the function level at the customer site.

1.2 Migration from Db2 11 to Db2 12

Db2 12 is adopting a continuous delivery step approach to delivering new functions. This approach involves shipping functional enhancements and service (both preventative service and defect fixes) in the same service stream. By enabling customers to adopt new function and service in a controlled manner, the Db2 development team is giving customers the confidence to continue to move forward with maintenance in their normal scheduling scheme. Continuous delivery also gives customers the flexibility needed to run their businesses while ensuring the stability of their systems. Using this approach, customers can activate new functions when they want.

Note: You can only migrate to Db2 12 from Db2 11 new-function mode (NFM).

1.2.1 Function levels

Since Version 8, Db2 has used function levels to identify when new features are available for use. The capability of the system was externalized in the RIBRELX and through various other externals, such as via `GET_VARIABLE(SYSIBM.VERSION)`. If you look at the RIBRELX control structure, there are 2 bytes of reserved space. Db2 12 extends the simple function level that was used in the past by using the additional 2 bytes of reserved space.

Figure 1-1 shows more information about RIBRELX.

- Db2 already has a function level that we expose to applications
 - The RIBRELX is exposed via various interfaces and is available to applications after a “CONNECT” statement. For instance the following info is available after a remote connect:
 - Database Connection Information
 - Database server = DB2 z/OS 12.1.5
 - SQL authorization ID = SYSADM
 - Local database alias = DB2A
- RIBRELX values is defined as VV.R.M where M is the Mod level - Since Db2 8:
 - Mod Levels 1-4 have been reserved for CM (New Function not Available for Db2 12)
 - Mod Levels 5-9 have been reserved for NFM (New Function Available for Db2 12)
- Solution
 - Extend RIBRELX
 - Mod levels 100, 200, 300, and 400 will be New Function not Available
 - Mod level 500-999 will be New Function Available

Figure 1-1 Extending RIBRELX

In Db2 12, RIBRELX is extended for the function levels based on the existing concept of a .1 being equivalent to conversion mode (CM) and .5 being NFM that was used in previous releases. Db2 12 at function level V12R1M100 is similar to CM in previous releases. At function level V12R1M100, a Db2 12 system can coexist with Db2 11 or can fall back to Db2 11. After V12R1M500 is activated, coexistence or fall back to Db2 11 is no longer possible.

Db2 12 introduces a new concept of function levels in the continuous delivery model to manage groups of new enhancements and capabilities. A *function level* enables a particular set of new Db2 capabilities and enhancements that were delivered (via PTF) and applied in a single continuous stream of Db2 code. This single continuous stream can include support for new capabilities, defect fixes, and preventative service items. The new capabilities and enhancements are not active when they are first applied to the system. Before the new capabilities of a function level can be used, that function level must be activated using the **-ACTIVATE** command. In addition, before an application can take advantage of new capabilities, the application must also specify the corresponding application compatibility (APPLCOMPAT) value.

Enhancements to Db2 are enabled for use when you use the **-Activate Function Level** command. Each function level corresponds to a single APAR that enables a set of enhancements that were previously delivered in the service stream. Before you can activate a function level, your data sharing group or subsystem must be at the appropriate catalog level (if applicable) and every member must be at the required code level for that function level. A particular function level might enable one or more enhancements. *Activation of a function*

level also activates the enhancements and capabilities that are introduced by all lower function levels.

Function levels are specified by the 9-byte strings that correspond to the Db2 version, release, and modification value, in the following format:

VvvRrrMmmm

In this format, *vv* is the version, *rr* is the release, and *mmm* is the modification level.

Db2 12 general availability code supports the following function levels:

► V12R1M100

This string identifies the function level after migration to Db2 12 (function level 100). Consider function level 100 as being similar to a CM in prior releases of Db2.

► V12R1M500

As the Db2 development team delivers more function levels in the continuous stream of Db2 code, the modification level, *mmm*, in the function level string with value of 500 or greater is reserved for the various function levels where new enhancements are supported. This function level can be thought of as similar to NFM in Db2 11.

► V12R1M501

Db2 has already shipped the first function level beyond base Db2 12 capability. V12R1M501 added support for the LISTAGG function. After function level V12R1M501 has been activated, applications that want to use LISTAGG must be bound with **APPLCOMPAT(V12R1M501)** to enable the application to use the new SQL capability.

► V12R1M100* (M100 star) and V12R1M500* (M500 star)

The *mode indicates that you were at a higher level. When in M100, you cannot use any new Db2 12 function. When you go to M500, you can start to use Db2 12 function.

Let's use partition-by-range table spaces with relative page numbering (PBR RPN) table spaces as an example. You have moved to M500 and created a PBR RPN table space. If you go back to M100*, you can still use the new PBR RPN table space, but you cannot create any new PBR RPN table spaces in any application not already bound at APPLCOMPAT(V12R1M500), and you cannot BIND any new applications with APPLCOMPAT(V12R1M500).

Similarly in M500*, an application bound with APPLCOMPAT(V12R1M501) can use the LISTAGG function, but new packages cannot be bound (with the **BIND** command) to exploit the LISTAGG functions. However, applications already bound with APPLCOMPAT(V12R1M501) can be rebound (using the **REBIND** command).

To activate a new function level and start to use it, use the **-ACTIVATE** command. On successful completion of the command, all function for that specified level and any previous levels become available. It is important to understand that there is not a one-to-one mapping between function levels and individual enhancements. A function level consists of individual items that have been shipped in a dormant state and are enabled by the **-ACTIVATE** command. Use the **-ACTIVATE** command to move forward to a function level and if you need to go back to a previous level use the same command with a previous function level to return to the prior function level.

A customer might want to go back to a previous mode level for various reasons. Going to a previous mode or * (asterisk) mode level is similar to CM* that you can return to after enabling new function mode (ENFM) or NFM in previous releases. The reason a customer returns to a previous level depends on what function is delivered in that level.

Let's say the Db2 development team has a new type of index (for example, type 3 indexes). Going to function level *XXX* enables a type 3 index to be created. Going to a previous level (a * function level) prevents new type 3 indexes from being created but still allows existing type 3 indexes to be accessed.

The following examples demonstrate use of the **-ACTIVATE** command to test and activate Db2 12 function levels.

Testing function level activation

The following **-ACTIVATE** command checks whether the subsystem or data sharing group is ready for the specified function level:

```
-ACTIVATE FUNCTION LEVEL(V12R1M500) TEST
```

The DSNU757I message (shown in Example 1-1) indicates that the group is ready for the specified level. Because TEST is specified in the command, 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.

Example 1-1 Message stating group is ready for specified level

```
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
```

Activating a function level

The following **-ACTIVATE** command attempts to activate the specified function level:

```
-ACTIVATE FUNCTION LEVEL (V12R1M500)
```

The DSNU757I message (shown in Example 1-2) indicates that function level 500 is successfully activated for that data sharing group.

Example 1-2 Message indicates function level 500 successfully activated

```
DSNU757I  -DB2A DSNUGCCA
*** BEGIN ACTIVATE FUNCTION LEVEL (V12R1M500)
          FUNCTION LEVEL (V12R1M500) SUCCESSFULLY ACTIVATED
          CATALOG LEVEL(V12R1M500)
          CURRENT FUNCTION LEVEL(V12R1M500)
          HIGHEST ACTIVATED FUNCTION LEVEL(V12R1M500)
          HIGHEST POSSIBLE FUNCTION LEVEL(V12R1M500)
DSN9022I  -DB2A DSNZACMD '-ACTIVATE FUNC' NORMAL COMPLETION
```

Attempting to activate a function level with ineligible members

The following **-ACTIVATE** command attempts to activate function level 500:

```
-ACTIVATE FUNCTION LEVEL (V12R1M500)
```

The DSNU757I message (shown in Example 1-3) indicates that the group is not yet ready for the activation of function level 500. The example subsystem is being migrated from Db2 11. One or more members are not at the required code level.

Example 1-3 Message indicates the group is not ready for activation to function level 500

```
DSNU757I  -DB2A DSNUGCCA
*** BEGIN ACTIVATE FUNCTION LEVEL (V12R1M500)
          GROUP NOT ELIGIBLE FOR FUNCTION LEVEL (V12R1M500)
          MEMBER(S) NOT STARTED WITH REQUIRED CODE LEVEL
          CATALOG LEVEL(V12R1M500)
          CURRENT FUNCTION LEVEL(V12R1M100)
          HIGHEST ACTIVATED FUNCTION LEVEL(V12R1M100)
          HIGHEST POSSIBLE FUNCTION LEVEL(V12R1M100)
```

DB2		CURRENT		CAPABLE FUNCTION LEVELS	
MEMBER	ID	CODE-LEVEL	LOWEST	HIGHEST	STATUS
DB2A	1	V12R1M500	V12R1M100	V12R1M500	ELIGIBLE
DB2B	2	V11R1M500	V11R1M500	V12R1M100	NOT ELIGIBLE
DB2C	3	V12R1M500	V12R1M100	V12R1M500	ELIGIBLE

```
DSN9022I  -DB2A DSNZACMD '-ACTIVATE FUNC' NORMAL COMPLETION
```

Activating a lower (*) function level

Assuming that the subsystem or data sharing group is at function level 500, the following **-ACTIVATE** command reverts the subsystem or group to function level 100*:

```
-ACTIVATE FUNCTION LEVEL (V12R1M100)
```

The DSNU757I message (shown in Example 1-4) indicates that function level 100* is activated.

Example 1-4 Message indicates that function level 100 is activated*

```
DSNU757I  -DB2A DSNUGCCA
*** BEGIN ACTIVATE FUNCTION LEVEL (V12R1M100)
          FUNCTION LEVEL (V12R1M100) SUCCESSFULLY ACTIVATED
          CATALOG LEVEL(V12R1M500)
          CURRENT FUNCTION LEVEL(V12R1M100*)
          HIGHEST ACTIVATED FUNCTION LEVEL(V12R1M500)
          HIGHEST POSSIBLE FUNCTION LEVEL(V12R1M500)
DSN9022I  -DB2A DSNZACMD '-ACTIVATE FUNC' NORMAL COMPLETION
```

1.2.2 Relationship between code levels and function levels

Keep in mind the following considerations that affect the relationship between maintenance and function levels going forward:

- ▶ The function level that a Db2 member is capable of running is directly related to the code or maintenance level of that member.
- ▶ The maintenance level of a Db2 member is associated with one or more PTFs.
- ▶ An RSU level (occurring monthly or quarterly) might or might not contain PTFs for one or more function levels.
- ▶ In data sharing, all members must be capable of supporting a function level before that function level can be activated.
- ▶ Each member has a function level capability, meaning the code running on that member is capable of supporting a specific function level.
- ▶ Maintenance is Db2 member scope, that is members can run code capable of supporting different function levels.
- ▶ Function level activation is group scope. After a function level is active, all members that are already part of that group or new members joining that group must be capable of supporting that function level.
- ▶ Although you can return to a previous * function level, doing so does not change the required maintenance level for a group. All members of the group must remain capable of supporting the highest activated function level.

1.2.3 Relationship between a function level and APPLCOMPAT

The APPLCOMPAT level of a package is directly related to the function level of the group. The following items explain the relationship between a function level and application compatibility:

- ▶ You cannot **BIND** (or set in any other way) an APPLCOMPAT value that is higher than the current function level of the group.
- ▶ APPLCOMPAT controls Data Manipulation Language (DML) (SQL and XML), Data Definition Language (DDL) (such as CREATE and ALTER), and Data Control Language (DCL) (GRANT and REVOKE) capability in an application (package). For a function that requires a specific APPLCOMPAT value to be used, the function level for the group must be at the corresponding function level or higher.
- ▶ The default APPLCOMPAT level for new **BIND** commands is controlled by the APPLCOMPAT subsystem parameter.
- ▶ The default APPLCOMPAT level for **REBIND** is the value currently in the catalog; however, it can be changed by explicitly specifying a value on the **REBIND** command. It is a sticky option only after the APPLCOMPAT columns are populated.
- ▶ There is no urgent need to change the APPLCOMPAT level for an application that has already been deployed. An existing application that has already been tested and deployed should not need anything from a new APPLCOMPAT level.

1.2.4 Catalog levels

There always has been a direct relationship between the Db2 catalog and Db2 code. Traditionally, the catalog and code are in sync. The Db2 12 migration is an example of this. When you start Db2 12, you must run the **CATMAINT** utility control statement before doing anything else. The introduction of the function level concept in Db2 12 softens this relationship in the following ways:

- ▶ A catalog change is not required until the function level that requires that change is activated.
- ▶ A catalog change might require a certain lower function level be active before the catalog change occurs (that is, preconditioning).
- ▶ A catalog level requires all members be at the code level that supports the change (this is similar to function level activation). The **CATMAINT** utility job checks to make sure that all members are capable of supporting the objects that will be created or modified as part of the **CATMAINT** job.
- ▶ This implementation might seem familiar to CM, ENFM and NFM in the sense that the concepts are similar:
 - Running with new maintenance is similar to running in CM, that is you can back off the maintenance level and you can coexist with prior maintenance levels. The backout of code in CM was fallback to the previous release.
 - Running the **CATMAINT** utility prevents backing off any maintenance that is required by that catalog level, this is similar to ENFM in the past (you cannot fall back to or coexist with maintenance that does not support the newly created or modified objects).
- ▶ A **CATMAINT** run is cumulative. When you run the **CATMAINT** utility with a specific level, it applies all changes to the catalog for all levels up to and including the specified level.
- ▶ After the **CATMAINT** run is complete, you might have new objects (tables or indexes, for example) or columns defined on Db2 catalog tables. However, the new objects or columns will not be actively used until the function level that requires them have been activated.
- ▶ The **-ACTIVATE FUNCTION LEVEL** command enables usage of new function (this is similar to NFM).
- ▶ Actual usage of the new catalog tables or fields might or might not require binding an application with a new APPLCOMPAT value.

1.2.5 Migrating Db2 11 data sharing groups to Db2 12

Figure 1-2 shows a timeline (from left to right) for how a Db2 11 data sharing group can be migrated to Db2 12 function level V12R1M100, in coexistence mode.

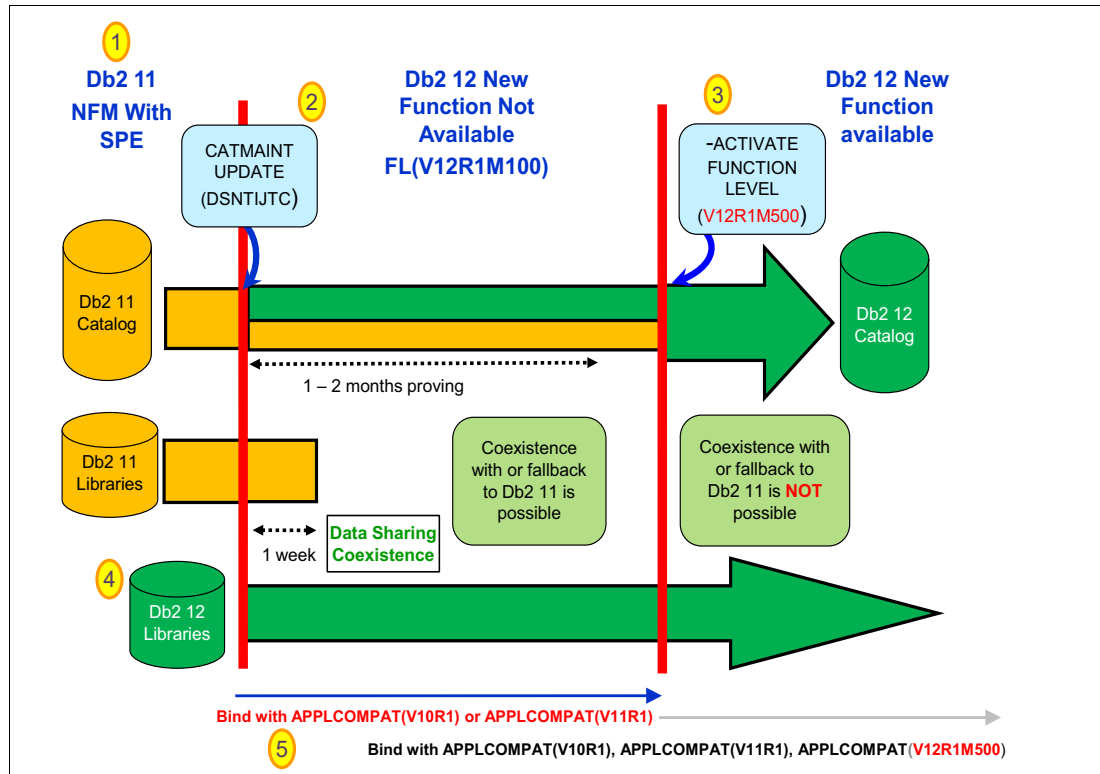


Figure 1-2 Migrate Db2 11 to Db2 12

As shown in Figure 1-2, this scenario is running Db2 V11 (Figure 1-2 item 1) and starts a data sharing group member with Db2 12 libraries (Figure 1-2 item 4). When that member is started, the **CATMAINT UPDATE** statement is used on the Db2 member to convert the Db2 11 catalog to Db2 12 (Figure 1-2 item 2). After a period of time all members are running Db2 12.

When the customer is sure there is no longer a need to fall back to or coexist with Db2 11, the **-ACTIVATE FUNCTION LEVEL (V12R1M500)** command is issued (Figure 1-2 item 3) to enable new feature use (Db2 12 capabilities). After activation of function level M500, applications that want to use Db2 12 capability (such as, SQL, DDL, and DML) can be rebound with the **BIND APPLCOMPAT (V12R1M500)** command (Figure 1-2 item 5).

Customers typically run in data sharing coexistence mode for a few hours to a few weeks while the new release is proven. Until the **-ACTIVATE FUNCTION LEVEL(V12R1M500)** command runs, you can only use **APPLCOMPAT(V10R1)** or **(V11R1)**, which ensures that any application you create during that time period can be executed on a down level member, if needed.

After the **-ACTIVATE FUNCTION LEVEL(V12R1M500)** command runs, bind with **APPLCOMPAT(V10R1)** or **(V11R1)** or **(V12R1M500)**. At that point, you can actually start using new SQL, XML, DDL, and other syntax and capabilities of V12.

Important: On successful completion of the **-ACTIVATE** command, all function for the specified level and any previous levels becomes available. For migration to Db2 12 from Db2 11, the level of V12R1M100 (new function not available) and V12R1M500 (new function available) are the base function levels. After V12R1M500 is adopted, coexistence with and fallback to Db2 11 is not possible. Db2 11 will not start and a Db2 11 member cannot join a data sharing group once function level M500 has been activated. Use of this command requires that the data sharing group contains no active Db2 11 members.

The **-ACTIVATE** command can be issued from a z/OS console, a DSN session under TSO, a Db2I panel (Db2 COMMANDS), an IBM IMS™ or IBM CICS® terminal, or a program using the instrumentation facility interface (IFI). To execute the **-ACTIVATE** command, you must use a privilege set that includes installation SYSADM or installation SYSOPR authority. The installation SYSOPR authority enables you to install or migrate Db2 without access to user objects.

For more information about the **-ACTIVATE** command, refer to *IBM DB2 12 for z/OS Technical Overview*, [SG24-8383](#) or [IBM Knowledge Center](#).

1.3 New function activation beyond base Db2 12 capability

This section provides a few examples of function activation that helps demonstrate how to activate function levels beyond V12R1M500.

1.3.1 Activate new function without catalog changes

Figure 1-3 shows activation of function level V12R1M501. This function level shipped about 6 months after Db2 12 GA.

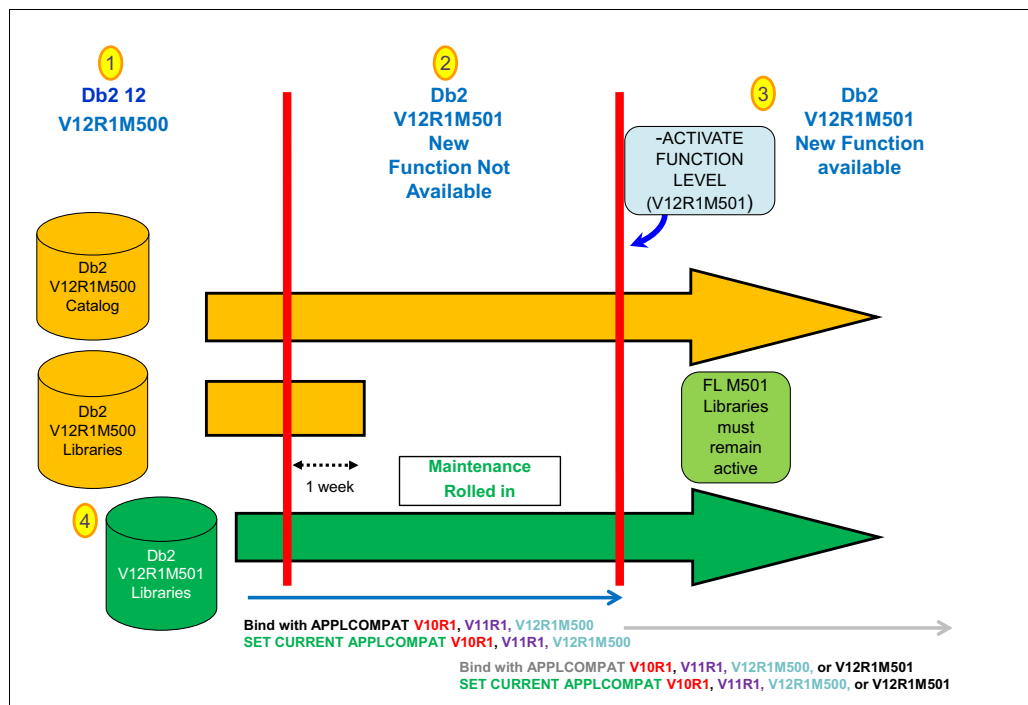


Figure 1-3 New function activation without catalog changes

Typically, customers apply corrective service two to four times a year. In this scenario (shown in Figure 1-3), the customer is running Db2 V12R1M500 (Figure 1-3 item 1). A new library labeled *Db2 V12R1M501 Libraries* is installed (Figure 1-3 item 4). In this case, the library supports a new code level M501 that is rolled around to all the members of the data sharing group (Figure 1-3 item 2). After the customer determines that the maintenance level is stable, function level M501 is activated using the **-ACTIVATE** command.

Until you activate M501, you can only apply BIND or REBIND with **APPLCOMPAT** V10R1, V11R1, or V12R1M500. After you execute **-ACTIVATE FUNCTION LEVEL V12R1M501** (Figure 1-3 item 3), you can bind or rebind applications to V12R1M501 also and use new function or SQL capabilities that are delivered as part of V12R1M501. In the case of M501, the LISTAGG built-in function was added and applications must be bound with V12R1M501 to use this new SQL function.

1.3.2 Activate new function with catalog changes

Figure 1-4 shows a hypothetical new function activation with catalog changes involved.

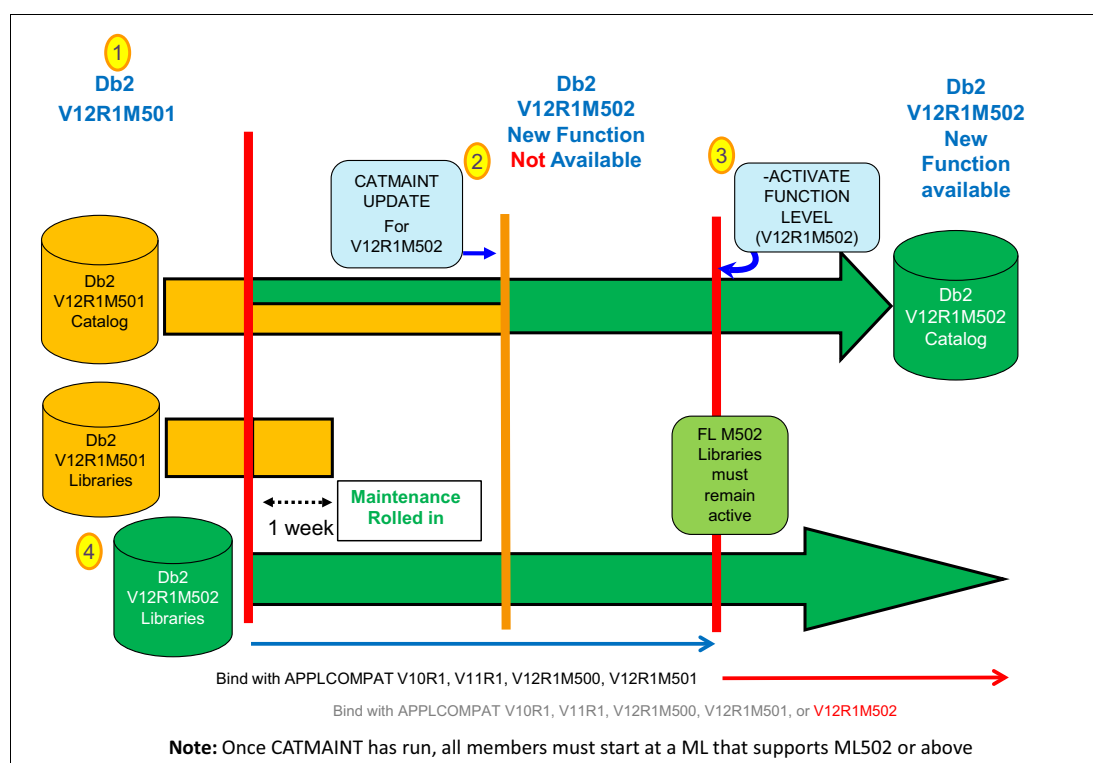


Figure 1-4 Hypothetical new function activation with catalog changes

In this scenario (shown in Figure 1-4), the customer is at function level V12R1M501 (Figure 1-4 item 1). The objective of the scenarios is to show moving to a new function level with one or more new capabilities involving catalog schema changes. The first task is to start Db2 with a library capable of supporting V12R1M502 (Figure 1-4 item 4). The modification level is rolled around to the members of the data sharing group. The new library contains all the PTFs necessary to activate the new function level (V12R01M502). After the maintenance level is stable, the customer needs to run the **CATMAINT** utility prior to activation of function level M502.

Figure 1-4 item 2 shows the execution of the **CATMAINT UPDATE** command. The **CATMAINT UPDATE** command can be run any time before activation of function level V12R1M502. As

discussed in 1.3.1, “Activate new function without catalog changes” on page 14, after M502 is activated, its functionality can be used.

If the **-ACTIVATE FUNCTION LEVEL** command with the **TEST** option was executed, it tells whether or not to run the **CATMAINT UPDATE** command. For an example of the message, see “Testing function level activation” on page 9. In this case the **-ACTIVATE FUNCTION LEVEL** command was executed (Figure 1-4 item 3).

With the **CATMAINT UPDATE** command, either new objects are created or existing objects are altered. After the **CATMAINT** utility executes, all members of the data sharing group must run maintenance that is capable of understanding the changed catalog. As such, make sure you are satisfied with your maintenance level before you execute the **CATMAINT** utility, because the command locks in the new maintenance level.

1.3.3 Activate new function with catalog change while skipping function levels

Figure 1-5 shows a hypothetical new function activation with catalog changes while skipping function levels.

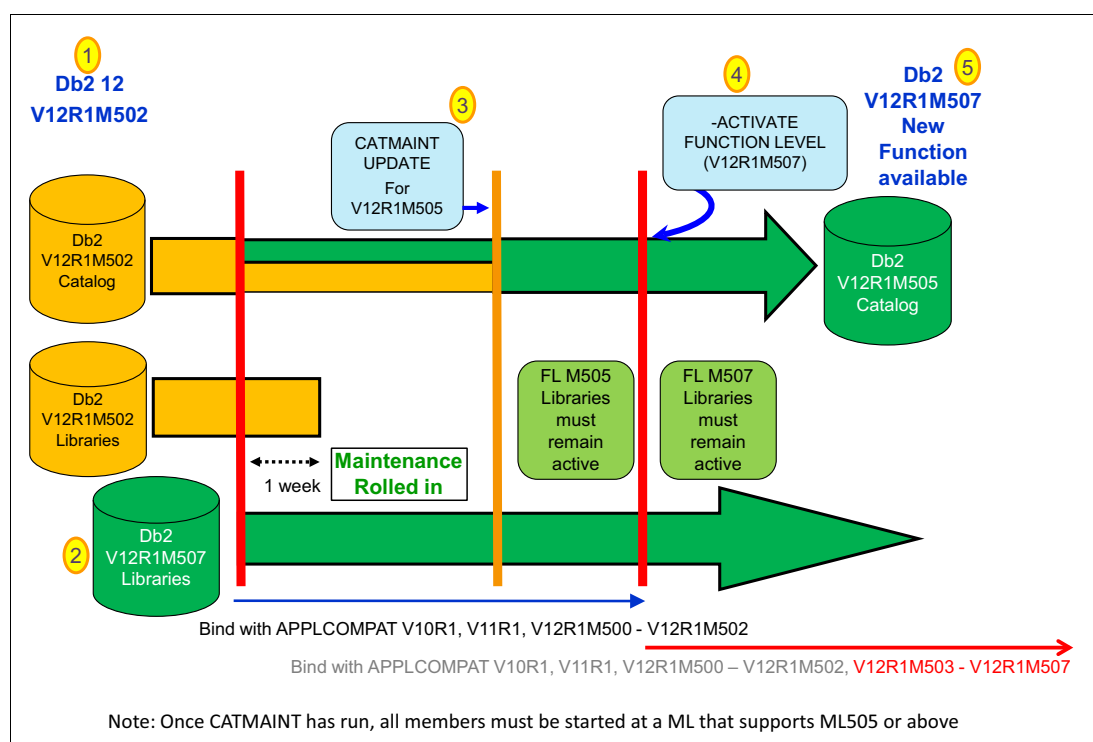


Figure 1-5 Hypothetical new function activation with catalog changes and skipping function levels

In this scenario (shown in Figure 1-5), the existing Db2 code library is at V12R1M502 (Figure 1-5 item 1). As part of a maintenance cycle, maintenance capable of supporting Db2 V12R1M507 is applied starting with one member (Figure 1-5 item 2) of the data sharing group. After a period of time, all members are running the new maintenance. When ready, the **CATMAINT** utility (Figure 1-5 item 3) is run that is required for function level V12R1M505. This **CATMAINT** utility is required prior to activating function level V12R1M507. After the **CATMAINT** run is complete, you likely would immediately execute the **-ACTIVATE FUNCTION LEVEL (V12R1M507)** command (Figure 1-5 item 4), because the goal was to activate the higher level functional capability. After the **-ACTIVATE** command is issued, Db2 V12R1M507 (Figure 1-5 item 5) capability is available across the environment.

You can choose to enable new capabilities in applications on a package-by-package basis, as necessary for applications. It is expected that not every application is modified to run with every new function level adopted by a customer. Rather, only those applications that need to use a specific new capability are modified to adopt a new APPLCOMPAT level. Any capability not enabled by APPLCOMPAT, for example the new UTILITY capability or syntax, are available immediately after activation of the function level.

In general, customers will not expose applications to new capability as they adopt new function levels. Essentially the APPLCOMPAT bind option provides an island of stability for an application. Applications are developed, tested, and deployed at a specific APPLCOMPAT level and that APPLCOMPAT level provides a stable execution environment for that application as different function levels are activated by a customer. It is expected that applications will adopt new APPLCOMPAT levels when they need to use new Db2 capability as part of a new development, test, and deployment cycle. Db2 12 supports Db2 11 and Db2 10 application compatibility (APPLCOMPAT(V10R1) and APPLCOMPAT(V11R1)), which demonstrates IBM commitment to support stable application execution environments over long periods of time for customers.

1.3.4 Function activation of previous level (* mode)

Db2 11 introduced the concept of application compatibility. Db2 11 has a Db2 10 mode **APPLCOMPAT (V10R1)** and a Db2 11 mode **APPLCOMPAT (V11R1)**. Db2 12 adds a Db2 12 mode **APPLCOMPAT (V12R1M500)**. After you have activated function level V12R1M500, you can bind applications with **APPLCOMPAT (V12R1M500)** and start using Db2 12 capabilities that are delivered in the base product. When Db2 customers adopt Db2 12, they have the opportunity to adopt additional capability that Db2 shipped in function levels beyond V12R1M500. For example, function level V12R1M501 already shipped and is available with APAR PI70535.

Figure 1-6 shows that the values you can specify for **APPLCOMPAT** are limited by the function level of your system. With one exception, that is discussed in the following content, you can only specify an **APPLCOMPAT** value that is less than or equal to your current function level.

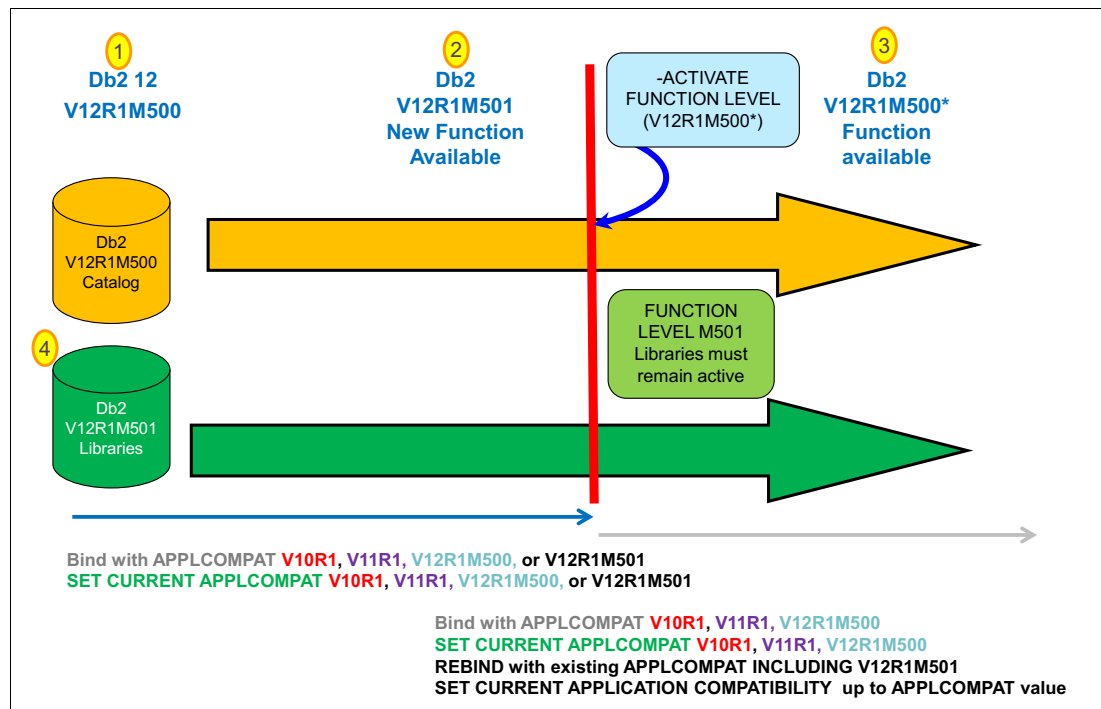


Figure 1-6 Function activation of previous level (* mode)

In this scenario (shown in Figure 1-6), the Db2 Catalog is at the V12R01M500 level (Figure 1-6 item 1), and the code library can support a function level of M501 (Figure 1-6 item 4). Function level of M501 has been activated (Figure 1-6 item 2). An application is bound with **APPLCOMPAT (V12R1M501)**. For some reason the customer wants to fall back to a lower or previous function level. The **-ACTIVATE FUNCTION LEVEL (V12R1M500)** command can be executed to take the data sharing group or subsystem (if non data sharing) from M501 back to M500* (Figure 1-6 item 3).

After the function level is activated to the previous level, you cannot BIND new applications with **APPLCOMPAT (V12R1M501)**. However, you can REBIND existing applications with **APPLCOMPAT (V12R1M501)**. Additionally, applications already bound with **APPLCOMPAT (V12R1M501)** continue to allow LISTAGG to be used. Newly bound applications are limited to **APPLCOMPAT (V12R1M500)** or below.

The special register CURRENT APPLICATION COMPATIBILITY can be set dynamically to any level equal to or below the APPLCOMPAT bind option for that package. For example, a package bound with **APPLCOMPAT (V12R1M501)** can set the CURRENT APPLICATION COMPATIBILITY special register to V10R1, V11R1, V12R1, V12R1M500 or V12R1M501 while another package bound with **APPLCOMPAT (V12R1M500)** is limited to setting the special register to V12R1M500 or below. An application might want to dynamically change the CURRENT APPLICATION COMPATIBILITY special register to enable a previous level because that was the level the application was tested and deployed with or possibly to avoid an incompatible change made by Db2 at a higher level.

Activating function levels

When first starting a Db2 12 system that was migrated from Db2 11, the current active function level is V12R1M100. This is also true for the data sharing group, if there is a Db2 11 coexistent member.

The **-ACTIVATE FUNCTION LEVEL** command is used to activate a function level which could be either higher or lower than the current function level after all data sharing members are on Db2 12. The **-ACTIVATE FUNCTION LEVEL** command checks all required actions and pre-requisites before the command is successful.

To activate function level V12R1M500 for the new functions shipped at Db2 12 GA, all members of the data sharing group must be at a Db2 12 code level and the catalog must already be converted to the V12R1M500 level (the catalog for Db2 12 is updated on initial migration so even when Db2 12 is at level V12R1M100, the catalog level is V12R1M500). After function level V12R1M500 is activated, a Db2 11 member cannot be started.

The scope of the **-ACTIVATE FUNCTION LEVEL** command is GROUP. If the specified function level is successfully activated, a successful message DSNU757I is displayed. A row is inserted into the SYSIBM.SYSLEVELUPDATES catalog table to indicate that the activation completed successfully.

If unsuccessful, the same DSNU757I message is issued, indicating the reason the group or Db2 subsystem could not be taken to the specified function level.

The **-ACTIVATE FUNCTION LEVEL** command **TEST** option is useful to determine whether the Db2 subsystem or data sharing group meets all requirements for a function level to be activated. The function level is not changed when the **TEST** option is specified.

Figure 1-7 shows the execution of the **-ACTIVATE FUNCTION LEVEL** command for the V12R1M100 function level and the results which indicate the execution completed normally. In this case, a * mode activation (to a previous function level) because the system was at function level M500 when the command was issued.

```

DSN7100I -DB1A DSN7GCMD

-DB2A ACTIVATE FUNCTION LEVEL (V12R1M100)

*** BEGIN ACTIVATE FUNCTION LEVEL (V12R1M100)
FUNCTION LEVEL (V12R1M100) SUCCESSFULLY ACTIVATED
CATALOG LEVEL(V12R1M500)
CURRENT FUNCTION LEVEL(V12R1M100*)
HIGHEST ACTIVATED FUNCTION LEVEL(V12R1M500)
HIGHEST POSSIBLE FUNCTION LEVEL(V12R1M500)

DSN9022I -DB2A DSNZACMD '-ACTIVATE FUNC' NORMAL COMPLETION

```

Figure 1-7 Activate Function Level M100

1.3.5 SYSLIB.SYSLEVELUPDATES table

Db2 12 introduces the SYSIBM.SYSLEVELUPDATES catalog table to help track the history of various changes to code level, catalog level, and function level over time. Each successful execution of the **-ACTIVATE** command is recorded in the SYSIBM.SYSLEVELUPDATES table so history of these commands can be reviewed. When a **CATMAINT UPDATE LEVEL** utility is run to change the catalog level, a row is also inserted in this table. In addition, the table records when the code level is updated for the Db2 subsystem or members in the data sharing group.

Table 1-1 shows the columns defined in the SYSIBM.SYSLEVELUPDATES catalog table.

Table 1-1 The SYSIBM.SYSLEVELUPDATES catalog table

Column name	Data type	Description
FUNCTION_LVL	VARCHAR(10)	Function level when this record was inserted
PREV_FUNCTION_LVL	VARCHAR(10)	Previous function level
HIGH_FUNCTION_LVL	VARCHAR(10)	Highest activated function level
CATALOG_LVL	VARCHAR(10)	Catalog level when this record was inserted
OPERATION_TYPE	CHAR(1)	Type of operation: <ul style="list-style-type: none"> ▶ "C" for catalog change ▶ "F" for function level ▶ "M" for maintenance update
EFFECTIVE_TIME	TIMESTAMP(12)	Time when the operation completed
EFFECTIVE_LRSN	CHAR(10)	RBA (or LRSN for data sharing) depicting the time that an operation completed
OPERATION_TEXT	VARCHAR(256)	The text of the operation
GROUP_MEMBER	VARCHAR(24)	Name of the group member on which the operation was executed

The contents of the SYSIBM.SYSLEVELUPDATES table (shown in Figure 1-8) were returned by a SELECT statement. The returned contents include function levels, catalog levels, and code levels of the Db2 subsystem or data sharing group. In the Figure 1-8 example, you can see at effective time 2016-06-27 07:31:18 the system was changed from function level 100 to

function level 500. At effective time 2016-07-29 14:57:43 the system was changed from function level 500 back to function level 100 (M100*). Finally, at effective time 2016-07-29 15:06:47 the system was changed back to function level 500.

```

SELECT * FROM SYSIBM.SYSLEVELUPDATES

```

	FUNCTION_ LVL	PREV_FUNC TION_LV	HIGH_FUNC TION_LVL	CATALOG_ LVL	OPERA TION_ TYPE	EFFECTIVE_TIME
1_	V12R1M500	V12R1M100	V12R1M500	V12R1M500	M	2016-06-27 07:31:18
2_	V12R1M100	V12R1M500	V12R1M500	V12R1M500	F	2016-07-29 14:57:43
3_	V12R1M500	V12R1M100	V12R1M500	V12R1M500	F	2016-07-29 15:06:47

	EFFEC TIVE_ LRSN	OPERATION_TEXT	GROUP_ MEMBER
1_	...	CATMAINT PROCESSING - DB2DEV	
2_	...	-ACTIVATE FUNCTION LEVEL (V12R1M100)	
2_	...	-ACTIVATE FUNCTION LEVEL (V12R1M500)	

Figure 1-8 Select from SYSIBM.SYSLEVELUPDATES table

1.3.6 Additional table changes

Additional tables have changed in Db2 12. Table 1-2 shows the tables with new catalog columns. These columns were added to tables to ensure that either a specific function level or a time stamp were recorded when an action occurred.

Table 1-2 Tables with new catalog columns

Table name	Column name	Data type	Description
SYSACCELERATEDPACKAGES SYSDYNQRY SYSPACKCOPY SYSPACKAGE SYSQUERY	FNLEVEL	VARCHAR(10)	Function level of the package/query at the time the row was inserted
SYSCONTROLS SYSINDEXES SYSROUTINES SYSTABLES	REGENERATETS	TIMESTAMP(12)	Time when the object was regenerated or altered
SYSKEYS SYSENVIRONMENT	CREATEDTS	TIMESTAMP(12)	Time when the row was inserted

Example 1-5 shows an example query to find the catalog level and function level when table MY.TABLE1 was created. This action can be accomplished by joining SYSLEVELUPDATES and SYSTABLES.

Example 1-5 Query MY.TABLE1

```
SELECT LVU.FUNCTION_LVL, LVU.CATALOG_LVL, LVU.EFFECTIVE_TIME, LVU.OPERATION_TEXT
FROM SYSIBM.SYSTABLES AS TAB,SYSIBM.SYSLEVELUPDATES AS LVU
WHERE TAB.CREATOR = 'MY' AND TAB.NAME = 'TABLE1'
AND TAB.CREATEDTS > LVU.EFFECTIVE_TIME
ORDER BY LVU.EFFECTIVE_TIME DESC
FETCH FIRST 1 ROWS ONLY
```

Figure 1-9 shows the results of that query.

FUNCTION_LVL	CATALOG_LVL	EFFECTIVE_TIME	OPERATION_TEXT
V12R1M503	V12R1M500	TIME 3	ACTIVATE FUNCTION LEVEL (V12R1M503)

Figure 1-9 MY.TABLE1 query results

Example 1-6 shows a query to find when function level M505 was activated.

Example 1-6 Query to find when M505 was activated

```
SELECT FROM SYSLEVELUPDATES AS LVU
WHERE LVU.FUNCTION_LVL = 'V12R1M505'
```

Figure 1-10 shows the results of the query in Example 1-6.

FUNCTION_LVL	PREV_FUNC_LVL	HIGH_FUNC_LVL	CATALOG_LVL	EFFECTIVE_TIME
V12R1M505	V12R1M500*	V12R1M505	V12R1M505	2016-04-20-12.38.03.706031000000

Figure 1-10 Query results from SYSLEVELUPDATES table

1.4 DISPLAY GROUP command

There are a few changes in the output of the **DISPLAY GROUP** command with Db2 12. Db2 11 shows the catalog level as a 3-byte string. Figure 1-11 shows the CATALOG LEVEL displayed as 111 (11.1). In Db2 12, the catalog level is shown in the new 9-byte string format, VvvRrrMmmm. The migration mode (displayed as *NEW FUNCTION(X)* in Figure 1-11 on page 22) is no longer displayed.

```

DSN7100I -DB1A DSN7GCMD
*** BEGIN DISPLAY OF GROUP(.....) CATALOG LEVEL(111)
      NEW FUNCTION(X) PROTOCOL LEVEL(2)
      GROUP ATTACH NAME(....)
-----
DB2      DB2 SYSTEM  IRLM
MEMBER   ID  SUBSYS  CMDPREF  STATUS  LVL NAME  SUBSYS  IRLMPROC
-----
DB1A     1  DB1A    -DB1A    ACTIVE  111 MVSA   DJ1A    DB1AIRLM
DB1B     2  DB1B    -DB1B    ACTIVE  111 MVSB   DJ1B    DB1BIRLM
DB1C     3  DB1C    -DB1C    ACTIVE  111 MVSC   DJ1C    DB1CIRLM
DB1D     4  DB1D    -DB1D    QUIESCED 111 MVSD   DJ1D    DB1DIRLM
-----
SCA  STRUCTURE SIZE:      1024 KB, STATUS= AC,   SCA IN USE:      11 %
LOCK1 STRUCTURE SIZE:      1536 KB
NUMBER LOCK ENTRIES:      262144
NUMBER LIST ENTRIES:      7353, LIST ENTRIES IN USE:      0
*** END DISPLAY OF GROUP(DSNDB10 )
DSN9022I -DB1A DSN7GCMD 'DISPLAY GROUP ' NORMAL COMPLETION

```

Figure 1-11 DISPLAY GROUP Db2 11

The output now always shows the following function levels:

- ▶ The current function level is a 9-byte string that indicates the *currently* activated function level.
- ▶ The highest activated function level is a 9-byte string that indicates *highest* function level that was *ever activated*.
- ▶ The highest possible function level is a 9-byte string that indicates the *highest* function level that *can be activated*.

The information can be used to determine the status of the subsystem or data sharing group prior to activating the next function level.

Db2 11 showed the individual Db2 code level as a 3-byte string. In Db2 12, the **DISPLAY GROUP** command shows the code level of each Db2 member in a 6-byte string format, which indicates version, release, and modification level (*vvrmmm*).

The examples in the following sections show various **DISPLAY GROUP** commands used throughout the migration process, migrating members in the data sharing to Db2 12 code one at a time, to activating a new function.

1.4.1 DISPLAY GROUP (coexistence): From Db2 12 member

Figure 1-12 shows a data sharing group with active, coexist members in both Db2 12 and Db2 11. Members DB2A, DB2B, and DB2C have migrated to Db2 12, so the catalog level is V12R1M500. The DB2 LVL column shows members at code levels of version 12, release 1, modification 500 and version 11, release 1, modification 500. Db2 11 is listed as 111500 in the DB2 LVL column for continuity with the Db2 12 information being displayed. A Db2 11 system must be in NFM to migrate to or coexist with Db2 12, so the level is shown as 111500.

Note: When first migrating to Db2 12, the code level is capable of both supporting V12R1M100 and V12R1M500. That is the reason why the code level under the DB2 LVL field is listed as *121500*.


```

DSN7100I -DB1A DSN7GCMDB
*** BEGIN DISPLAY OF GROUP(.....) CATALOG LEVEL(V12R1M500)
                                CURRENT FUNCTION LEVEL(V11R1M500)
                                HIGHEST ACTIVATED FUNCTION LEVEL(V11R1M500)
                                HIGHEST POSSIBLE FUNCTION LEVEL(V11R1M500)
                                PROTOCOL LEVEL(2)
                                GROUP ATTACH NAME(....)
-----
DB2      DB2      SYSTEM      IRLM
MEMBER   ID   SUBSYS  CMDPREF  STATUS  LVL   NAME      SUBSYS  IRLMPROC
-----
DB1A      1  DB1A   -DB1A   ACTIVE  121500 MVSA      DJ1A    DB1AIRLM
DB1B      2  DB1B   -DB1B   ACTIVE  121500 MVSB      DJ1B    DB1BIRLM
DB1C      3  DB1C   -DB1C   ACTIVE  121500 MVSC      DJ1C    DB1CIRLM
DB1D      4  DB1D   -DB1D   QUIESCED 111500 MVSD      DJ1D    DB1DIRLM
-----
SCA  STRUCTURE SIZE:      1024 KB, STATUS= AC,   SCA IN USE:      11 %
LOCK1 STRUCTURE SIZE:      1536 KB
NUMBER LOCK ENTRIES:      262144
NUMBER LIST ENTRIES:      7353, LIST ENTRIES IN USE:      0
*** END DISPLAY OF GROUP(DSNDB10 )
DSN9022I -DB1A DSN7GCMDB 'DISPLAY GROUP ' NORMAL COMPLETION

```

Figure 1-12 DISPLAY GROUP coexistence

1.4.2 DISPLAY GROUP (new function not available)

Figure 1-13 shows a data sharing group where all active members have been migrated to Db2 12 code level before activating any new function. The current function level is V12R1M100 and the highest possible function level is V12R1M500. There are two members in base maintenance capability and two members have upgraded to a maintenance capable supporting function level V12R1M503.

```

DSN7100I -DB1A DSN7GCMDB
*** BEGIN DISPLAY OF GROUP(.....) CATALOG LEVEL(V12R1M500)
                                CURRENT FUNCTION LEVEL(V12R1M100)
                                HIGHEST ACTIVATED FUNCTION LEVEL(V12R1M100)
                                HIGHEST POSSIBLE FUNCTION LEVEL(V12R1M500)
                                PROTOCOL LEVEL(2)
                                GROUP ATTACH NAME(....)
-----
Db2      Db2      SYSTEM      IRLM
MEMBER   ID   SUBSYS  CMDPREF  STATUS  LVL   NAME      SUBSYS  IRLMPROC
-----
DB1A      1  DB1A   -DB1A   ACTIVE  121500 MVSA      DJ1A    DB1AIRLM
DB1B      2  DB1B   -DB1B   ACTIVE  121500 MVSB      DJ1B    DB1BIRLM
DB1C      3  DB1C   -DB1C   ACTIVE  121503 MVSC      DJ1C    DB1CIRLM
DB1D      6  DB1D   -DB1D   ACTIVE  121503 MVSD      DJ1D    DB1DIRLM
-----
SCA  STRUCTURE SIZE:      1024 KB, STATUS= AC,   SCA IN USE:      11 %
LOCK1 STRUCTURE SIZE:      1536 KB
NUMBER LOCK ENTRIES:      262144
NUMBER LIST ENTRIES:      7353, LIST ENTRIES IN USE:      0
*** END DISPLAY OF GROUP(DSNDB10 )
DSN9022I -DB1A DSN7GCMDB 'DISPLAY GROUP ' NORMAL COMPLETION

```

Figure 1-13 DISPLAY GROUP new function not available

1.4.3 DISPLAY GROUP (New Function Active)

Figure 1-14 shows a data sharing group where all active members have been migrated to V12 and new function has been activated. In this example, the catalog is at M502. The current function level is at M502* (asterisk) and the highest activated function level is M503. In this case, the customer has gone from M503 back to M502 and the maintenance level is capable of supporting V12R1M505. You can see three of the members are at M505, and one of the members is capable of supporting M507. So it looks like this particular customer has advanced some code levels to M507 already.

```

DSN7100I -DB1A DSN7GCMD
*** BEGIN DISPLAY OF GROUP(.....) CATALOG LEVEL(V12R1M502)
                                CURRENT FUNCTION LEVEL(V12R1M502*)
                                HIGHEST ACTIVATED FUNCTION LEVEL(V12R1M503)
                                HIGHEST POSSIBLE FUNCTION LEVEL(V12R1M505)
                                PROTOCOL LEVEL(2)
                                GROUP ATTACH NAME(....)
-----
DB2          DB2      SYSTEM      IRLM
MEMBER      ID  SUBSYS  CMDPREF  STATUS  LVL   NAME      SUBSYS  IRLMPROC
-----
DB1A         1  DB1A   -DB1A   ACTIVE  121505 MVSA      DJ1A    DB1AIRLM
DB1B         2  DB1B   -DB1B   ACTIVE  121505 MVSB      DJ1B    DB1BIRLM
DB1C         3  DB1C   -DB1C   ACTIVE  121505 MVSC      DJ1C    DB1CIRLM
DB1D         4  DB1D   -DB1D   ACTIVE  121507 MVSD      DJ1D    DB1DIRLM
-----
SCA  STRUCTURE SIZE:      1024 KB, STATUS= AC,   SCA IN USE:      11 %
LOCK1 STRUCTURE SIZE:      1536 KB
NUMBER LOCK ENTRIES:      262144
NUMBER LIST ENTRIES:      7353, LIST ENTRIES IN USE:      0
*** END DISPLAY OF GROUP(DSNDB10 )
DSN9022I -DB1A DSN7GCMD 'DISPLAY GROUP ' NORMAL COMPLETION

```

Figure 1-14 DISPLAY GROUP new function available

1.4.4 DISPLAY GROUP (after CATMAINT)

Figure 1-15 is an example after a CATMAINT command runs. The catalog level actually exceeds the current function level. Catalog level is V12R1M506 and the current function level is V12R1M502* (star). The highest possible function level is V12R1M507. Notice that all the members are at the same code level and are capable of supporting function level M507. It is important to note that as soon as the catalog level became V12R1M506 the system can be started *only* with a maintenance that is capable of supporting function level M506 or above.

```

DSN7100I -DB1A DSN7GCMDB
*** BEGIN DISPLAY OF GROUP(.....) CATALOG LEVEL(V12R1M506)
                                     CURRENT FUNCTION LEVEL(V12R1M502*)
                                     HIGHEST ACTIVATED FUNCTION LEVEL(V12R1M503)
                                     HIGHEST POSSIBLE FUNCTION LEVEL(V12R1M507)
                                     PROTOCOL LEVEL(2)
                                     GROUP ATTACH NAME(....)
-----
DB2      DB2      SYSTEM      IRLM
MEMBER   ID   SUBSYS  CMDPREF  STATUS  LVL   NAME      SUBSYS  IRLMPROC
-----
DB1A     1   DB1A    -DB1A    ACTIVE  121507 MVSA      DJ1A    DB1AIRLM
DB1B     2   DB1B    -DB1B    ACTIVE  121507 MVSB      DJ1B    DB1BIRLM
DB1C     3   DB1C    -DB1C    ACTIVE  121507 MVSC      DJ1C    DB1CIRLM
DB1D     4   DB1D    -DB1D    ACTIVE  121507 MVSD      DJ1D    DB1DIRLM
-----
SCA  STRUCTURE SIZE:      1024 KB, STATUS= AC,   SCA IN USE:      11 %
LOCK1 STRUCTURE SIZE:      1536 KB
NUMBER LOCK ENTRIES:      262144
NUMBER LIST ENTRIES:      7353, LIST ENTRIES IN USE:      0
*** END DISPLAY OF GROUP(DSNDB10 )
DSN9022I -DB1A DSN7GCMDB 'DISPLAY GROUP ' NORMAL COMPLETION

```

Figure 1-15 DISPLAY GROUP (after CATMAINT)

1.5 Function level adoption preferred practices

The following function level adoption practices are preferred:

- ▶ PTFs (including RSUs) are applied using your standard maintenance procedures. The application of maintenance can increase the code level of a Db2 system.
- ▶ After a system is stable on maintenance, execute (if needed) the **CATMAINT** command.
After execution of the **CATMAINT** command, the system can be started only with a code level that supports the catalog.
- ▶ Issue the **-ACTIVATE** command. After it completes, the following items are true:
 - Function not related to SQL, DML, DCL syntax is available.
 - **BIND** or **REBIND** of packages with any **APPLCOMPAT** value pick up optimizer enhancements that are associated with the function level or other non-**APPLCOMPAT** related enhancements.
 - Non-stabilized dynamic SQL pick up optimizer enhancements or other non-**APPLCOMPAT** related enhancements.

1.6 APPLCOMPAT adoption preferred practices

After a new function level is considered stable allow new application feature rollout, as follows:

- ▶ **REBIND DBA** packages to allow new DDL to be used, which might be specific versions of SQL Processor Using File Input (SPUFI) or other admin type tools used by the DBA staff.
- ▶ **REBIND** application static packages with a higher **APPLCOMPAT** to use DDL/DML new functions and behaviors as needed by those applications.

- ▶ **REBIND** dynamic packages with a higher **APPLCOMPAT** to allow new SQL functions to be used as needed by those applications.
- ▶ **REBIND** distributed packages (** in separate collections) to allow new SQL functions to be used. Switch applications to use new distributed package collection.
- ▶ Use PLANMGMT extended:
 - **REBIND SWITCH (PREVIOUS)** to restore static to prior runtime structures.
 - **REBIND SWITCH (PREVIOUS)** for dynamic would restore prior **APPLCOMPAT**.
 - ** switching to prior **COLLID** for distributed dynamic would restore **APPLCOMPAT**.



Changes in Db2 12 affecting applications and more

This chapter focuses on Db2 12 support for SQL and application enablement for both local and remote applications. It includes the following topics:

- ▶ Application considerations
- ▶ New built-in global variables
- ▶ The IBM Data Server Driver

2.1 Application considerations

Db2 11 introduced the concept of *application compatibility*. The APPLCOMPAT bind option specifies the application compatibility behavior for static SQL statements in the package. It also sets the initial value of CURRENT APPLICATION COMPATIBILITY special register, which controls the application compatibility behavior for dynamic SQL statements in the package. The value of the APPLCOMPAT subsystem parameter specifies the default value if a value is not specified on the **BIND** command.

The APPLCOMPAT bind option is optional syntax for the **BIND PACKAGE**, **REBIND PACKAGE**, and **REBIND TRIGGER PACKAGE** commands. In addition, APPLCOMPAT was added as an option to the CREATE and ALTER statements for SQL Procedural Language (SQL PL), for either compiled or native, functions and procedures.

Option note: The APPLCOMPAT option was not added to the CREATE TRIGGER statement but was added to **REBIND TRIGGER PACKAGE** command. This change was done under the premise that when creating a trigger, the current SQL capability for the system (based on the APPLCOMPAT subsystem parameter) would be adopted. Use the **REBIND TRIGGER** command to control behavior for an existing trigger, either by maintaining current application behavior or by advancing the behavior to a newer level.

The format consists of version and release levels, with the two values supported in Db2 11 V10R1 and V11R1. The values were added for the following purposes:

- ▶ To separate incompatible behaviors introduced by a new release from release migration (migration of the system from Db2 10 to Db2 11). With the capabilities introduced in Db2 11, you can now migrate a Db2 system from one release to another without impacting application behavior.
- ▶ To enable new SQL functionality such as, new SQL syntax or semantics to be adopted in a controlled manner, application by application (at a package level), instead of subsystem wide, at the time of the system migration.

In Db2 12, the APPLCOMPAT bind option is extended to include the modification level that was introduced with the Function Level concept, in addition to Db2 10 and Db2 11 functionality. Support for the following specifications of APPLCOMPAT are available:

- ▶ V10R1: SQL behaviors are compatible with Db2 10.
- ▶ V11R1: SQL behaviors are compatible with Db2 11.
- ▶ V12R1M5xx: SQL behaviors are compatible with the capabilities of Db2 12 that are documented for the specified APPLCOMPAT level. In general, the APPLCOMPAT value cannot exceed the function level of the system. For example, if the function level of the system is M500, APPLCOMPAT(V12R1M501) cannot be specified. Db2 12 supports (at the time of this publication) the following Db2 12 APPLCOMPAT levels:
 - V12R1M500: This APPLCOMPAT level affects all DDL, DML, and DCL (GRANT, REVOKE, and TRANSFER) statements and enables base Db2 12 new functions (new capability that was delivered as part of Db2 12 at GA) in these areas. This capability is similar to the way that capability was enabled within these areas with new function mode (NFM) in Db2 10 and below, where all new capability for the release (DDL, DML, DCL, utility, and command syntax) was available once in new function mode (NFM).

Note that Db2 11 was slightly different because DML was controlled by APPLCOMPAT, but DDL and DCL were controlled by NFM.

- V12R1M501: This APPLCOMPAT level enables the LISTAGG SQL built-in function. LISTAGG was the only new function introduced with function level 501. Function level 501 was used by the Db2 Development team as a way to test the internal processes and procedures for function level development.

As new function levels are introduced, the corresponding new APPLCOMPAT values will be introduced for application usage (if necessary). Example 2-1 shows the **BIND PACKAGE** syntax with the APPLCOMPAT option.

Example 2-1 The BIND PACKAGE syntax with the APPLCOMPAT option

```
>>-BIND PACKAGE--(+-----+collection-id)----->
                        '-location-name.-'

>--+-----+----->
  '-APPLCOMPAT(+function-level+)-'
                +-V11R1-----+
                '-V10R1-----'
```

Before you can take advantage of new application related capabilities (such as new SQL or DDL syntax) that are delivered and enabled by a function level, you must also specify the corresponding application compatibility value for the package that uses that capability, through the **BIND** or **REBIND** command.

No changes are necessary to retrieve the APPLCOMPAT level of a package; the information is available in the same place as it was in Db2 11. The existing SYSIBM.SYSPACKAGE catalog table APPLCOMPAT column is VARCHAR(10) and is long enough to carry the extended APPLCOMPAT format added in Db2 12.

Note: Prior to Db2 12, the expectation was that customers migrate applications to a new release using APPLCOMPAT sometime during the course of the new release. For example, during the 3 or 4 years a customer was in production on Db2 11, most applications would be moved from APPLCOMPAT(V10R1) to APPLCOMPAT(V11R1). Two releases of compatibility were provided (Db2 12 supports V10R1 and V11R1) to enable skip release migration and to allow for some applications that are stable or possibly even being sunset and, thus, would not want to adopt new capability. This assumption was based on the then current 3 year release cycles for Db2.

With the adoption of continuous delivery with Db2 12, there is a recognition that not all customers will adopt every function level. In addition, some applications might need extended periods of stability, while other applications might be experiencing continuous delivery themselves and need access to the latest Db2 capabilities. Based on discussions with many customers, the Db2 development team believes that most applications will not adopt the current release base APPLCOMPAT value (for example, V12R1M500), but instead will move from a previous APPLCOMPAT level (such as V10R1 or V11R1) to an APPLCOMPAT level in Db2 12 that happens to be current at the time the application is deployed. The application then stays at that APPLCOMPAT level until the next time the application goes through a development, test, and deployment cycle.

2.1.1 The behavior of SQL, DDL, and DCL in Db2 12

One of the goals of Db2 12 is to provide islands of stability for applications to be developed, tested, and deployed. With the advent of continuous delivery, functional changes in Db2 come more frequently and with more granularity (smaller amounts of change, more often). This situation drives the need to control application stability at a finer level. In Db2 12, the

APPLCOMPAT bind option is extended to support function levels (for example, function level V12R1M501). Also in Db2 12, the APPLCOMPAT bind option is extended to support DDL and DCL, in addition to DML. Thus, to access Db2 12 functionality (for example, to create a new PBR RPN table space), you need to specify APPLCOMPAT (V12R1M500) or above. To use the new LISTAGG function, shipped with function level 501, you need to have an application bound with APPLCOMPAT (V12R1M501). When you adopt a new function level, which is a group-wide event, applications are not exposed to the changes introduced by that function level until they are rebound with the corresponding APPLCOMPAT for that function level. With this capability, applications are essentially stabilized at the functional API level that is associated with the APPLCOMPAT level under which they were developed, tested, and deployed.

Applications might be exposed to non-functional API behavioral changes when a function level is adopted. For example, in the past, when you migrated from Db2 10 to Db2 11 conversion mode (CM), your applications were exposed to new infrastructure changes in Db2. Similarly, when a new function level is adopted, your applications might be exposed to the changes that are associated with that function level. When and how exposed depends on the change and the application (Dynamic or Static).

For example, if you had dynamic SQL (or if you rebound static SQL) in CM mode (in for example Db2 11), your applications might have been exposed to new access paths. Access path changes affect the speed of processing an SQL statement, but they do not change the results of a query. In Db2 12, after you adopt function level V12R1M500, applications are potentially exposed to using fast traversal blocks (FTBs), even without rebinding. Using FTBs affects the code path inside Db2 and improves performance but does not affect the results of the query.

APPLCOMPAT capabilities: To provide applications as much stability as possible, it is important to understand and use the capabilities delivered in Db2 12 via APPLCOMPAT.

2.1.2 Dynamic SQL, DDL, and DCL

In Db2 11, by default, dynamic SQL executed under the APPLCOMPAT BIND option of the package used to execute the dynamic SQL. The SET CURRENT APPLICATION COMPATIBILITY statement assigns a value to the CURRENT APPLICATION COMPATIBILITY special register. This special register allows you to control the application compatibility level for dynamic SQL.

In Db2 11 it is possible to use the SET CURRENT APPLICATION COMPATIBILITY statement to set the CURRENT APPLICATION COMPATIBILITY special register to a level higher than your package was bound. For example, you can have the application compatibility level for SQL Processor Using File Input (SPUFI) set at V10R1 and use the SET CURRENT APPLICATION COMPATIBILITY statement to set the CURRENT APPLICATION COMPATIBILITY special register to V11R1 making it possible to access V11R1 SQL syntax.

In Db2 12, the APPLCOMPAT bind option controls the MAX value that can be specified by the SET CURRENT APPLICATION COMPATIBILITY statement. Thus, in Db2 12 you might only set the special register to a level that is equal to or below your bind option. For example, you can **BIND SPUFI** at APPLCOMPAT (V12R1M500). At this level, SPUFI has access to all Db2 12 functionality. However, you can set CURRENT APPLICATION COMPATIBILITY to V10R1 if you had an SQL statement you want to behave as though it was running in Db2 10. The key difference in Db2 12 is that the bind option is the maximum value you can use to set CURRENT APPLICATION COMPATIBILITY. For backward compatibility, you can still set the CURRENT APPLICATION COMPATIBILITY special register to V11R1 in a package that is bound with APPLCOMPAT (V10R1).

An APPLCOMPAT level remains active even if the function level is lowered to a * function level (for example, function level M501 moved to function level M500*). Although at a lower level, a

REBIND of an existing application retains the current (uplevel) APPLCOMPAT value even if the value exceeds the current * mode function level. The SET CURRENT APPLICATION COMPATIBILITY statement can be specified *up to* the value specified on the original **BIND** or **REBIND**.

2.2 New built-in global variables

To enable an API for the function levels introduced with continuous delivery in Db2 12, the built-in global variables described in the following sections were created. Because of single-phase migration, an application that retrieves these new variables can read these variables in Db2 11 NFM and Db2 12. This applies to any Db2 11 system that has applied the Db2 12 migration SPE and executed the associated CATMAINT statement.

2.2.1 PRODUCTID_EXT

The PRODUCTID_EXT global variable contains the extended product identifier of the database manager that was used to invoke the function. This global variable has the following characteristics:

- ▶ It is read only, with values maintained by the system.
- ▶ The type is VARCHAR(30).
- ▶ The schema is SYSIBM.
- ▶ The scope is session.

The format of the extended product identifier values is *pppvrrmmm*:

- ▶ *ppp* is a three-letter product code (such as, DSN for Db2).
- ▶ *vv* is the version.
- ▶ *rr* is the release.
- ▶ *mmm* is the modification level (such as, 100, 500, 501).

For example, *DSN1201500* identifies Db2 12 after the activation of Db2 12 new function (function level 500 or greater). If the variable is retrieved on a coexistent Db2 11 member of a data sharing group (which must be in NFM), the value DSN1101500 is returned.

2.2.2 CATALOG_LEVEL

The CATALOG_LEVEL global variable contains the current catalog level. This global variable has the following characteristics:

- ▶ It is read only, with values maintained by the system.
- ▶ The type is VARCHAR(30).
- ▶ The schema is SYSIBM.
- ▶ The scope is session.

The format of the catalog level values is *VvvRrMmmm*:

- ▶ *vv* is the version.
- ▶ *r* is the release.
- ▶ *mmm* is the modification level (such as 100, 500, 501).

For example, *V12R1M500* identifies Db2 12 after the activation of Db2 12 and the initial **CATMAINT** run for Db2 12 runs. If the variable is retrieved on a coexistent Db2 11 member of a data sharing group, the value V12R1M500 is returned after the initial **CATMAINT** run for Db2 12 runs on a Db2 12 member.

2.2.3 DEFAULT_SQLLEVEL

The DEFAULT_SQLLEVEL variable contains the default value of the SQLLEVEL SQL processing option (DECPSQLL). This global variable has the following characteristics:

- ▶ It is read only, with values maintained by the system.
- ▶ The type is VARCHAR(30).
- ▶ The schema is SYSIBM.
- ▶ The scope is session.

The format of the catalog level values is V10R1, V11R1, or VvvRrMmmm:

- ▶ vv is the version.
- ▶ r is the release.
- ▶ mmm is the modification level (such as 100, 500, 501).

For example, *V12R1M501* identifies function level 501.

2.3 The IBM Data Server Driver

The IBM Data Server Driver is a lightweight solution and the recommended preferred practice package for user or application server deployment. It provides robust runtime support for applications using ODBC, CLI, .NET, OLE DB, PHP, Ruby, JDBC, or SQLJ.

The application-function level that is used by applications employing the Data Server Driver is also controlled by the APPLCOMPAT bind option that is used for Data Server Driver packages. Starting with Db2 12 and with the latest Data Server Driver levels a collection name provided on the data source property specifies the Data Server Driver packages to be used.

Db2 12 controls the capabilities used by the Data Server Drivers by returning the APPLCOMPAT value of the package instead of the current function level of the Db2 system or data sharing group. This subtle change in behavior ensures that applications using the Data Server Drivers remain stable even while the Db2 function level advances. For example, Db2 12 using Data Server Driver packages bound with APPLCOMPAT(V11R1), will respond (on CONNECT) that the Db2 server is actually a Db2 11 server.

The CURRENT APPLICATION COMPATIBILITY special register can be set in the DSN_PROFILE tables but the value set must be less than or equal to the APPLCOMPAT bind option specified on the Data Server Driver packages that were loaded at connect time based on the COLLECTION specified.

2.3.1 The ClientApplCompat keyword

Any supported level of IBM Data Server Drivers should work with Db2 12 in function level M100 or M500, with an APPLCOMPAT(V12R1M500) and below.

Data Server Drivers must be at the following levels to use the Db2 12 application compatibility for V12R1M501 or greater in these situations:

- ▶ IBM Data Server Driver for JDBC and SQLJ: Versions 3.72 and 4.22, or later
- ▶ Other IBM data server clients and drivers: Db2 for Linux, UNIX, and Windows Version 11.1 Modification 1 Fix Pack or later

The ClientAppCompat keyword must be specified if the Data Server Driver packages are bound with APPLCOMPAT (V12R1M501) or higher. ClientAppCompat lets you control the capability of the client when updated drivers ship changes to enable new server capability. You might want to have specific control of driver capability in these instances:

- Db2 client driver introduces new behavior currently not controlled by Db2 application compatibility.
- Change needs to be controlled at the application level to ensure compatibility with new behavior.

The ClientAppCompat V12R1M500 keyword is required to access Db2 12 server capability that is shipped after general availability at function levels beyond Db2 V12R1M500.

Figure 2-1 shows the clientAppCompat keyword relationship to Db2 APPLCOMPAT.

clientAppCompat relationship to Db2 APPLCOMPAT			
Driver Level	clientAppCompat setting	APPLCOMPAT setting***	Application Capability
V10.5 FP2(-) V11 V11 FP1(+)	n/a n/a not provided	V10R1 V11R1 V12R1M100 V12R1M500	Applications may exploit Db2 10 functionality Applications may exploit Db2 11 functionality Applications may exploit Db2 11 functionality Applications may exploit Db2 V12R1M500 functionality
V10.5 FP2(-) V11 V11 FP1(+)	n/a n/a not provided	V12R1M501 – V12R1M5xx	SQLCODE -30025 Function Level V12R1M501 has a dependency on clientAppCompat being set
V11 FP1(+)	V12R1M500 To Exploit function level V12R1M501 and beyond, clientAppCompat must be set	V10R1 V11R1 V11R1M100 V12R1M501 V12R1M5xx – V12R1M5(zz-1)	Applications may exploit Db2 10 functionality Applications may exploit Db2 11 functionality Applications may exploit Db2 11 functionality Applications may exploit Db2 V12R1M501 functionality Applications may exploit Db2 V12R1M5xx functionality Applications may exploit Db2 V12R1M5(zz-1) functionality
V11 FP1(+)	V12R1M501(+)	V12R1M500	SQLCODE -30025 clientAppCompat must not exceed the server APPLCOMPAT setting
Vyy Fpy ###	V12R1M5zz	V12R1M5zz	Applications may exploit Db2 V12R1M5zz functionality Future function that requires new DRDA data flow
***This is the APPLCOMPAT setting for the IBM Data Server Driver Packages specified by the collection id ### Hypothetical future Driver fixpack needed to support a Db2 for z/OS server change at FL V12R1M5zz			

Figure 2-1 The clientAppCompat relationship to Db2 APPLCOMPAT

Figure 2-2 shows Data Server Drivers prior to Db2 11 fix pack 1*.

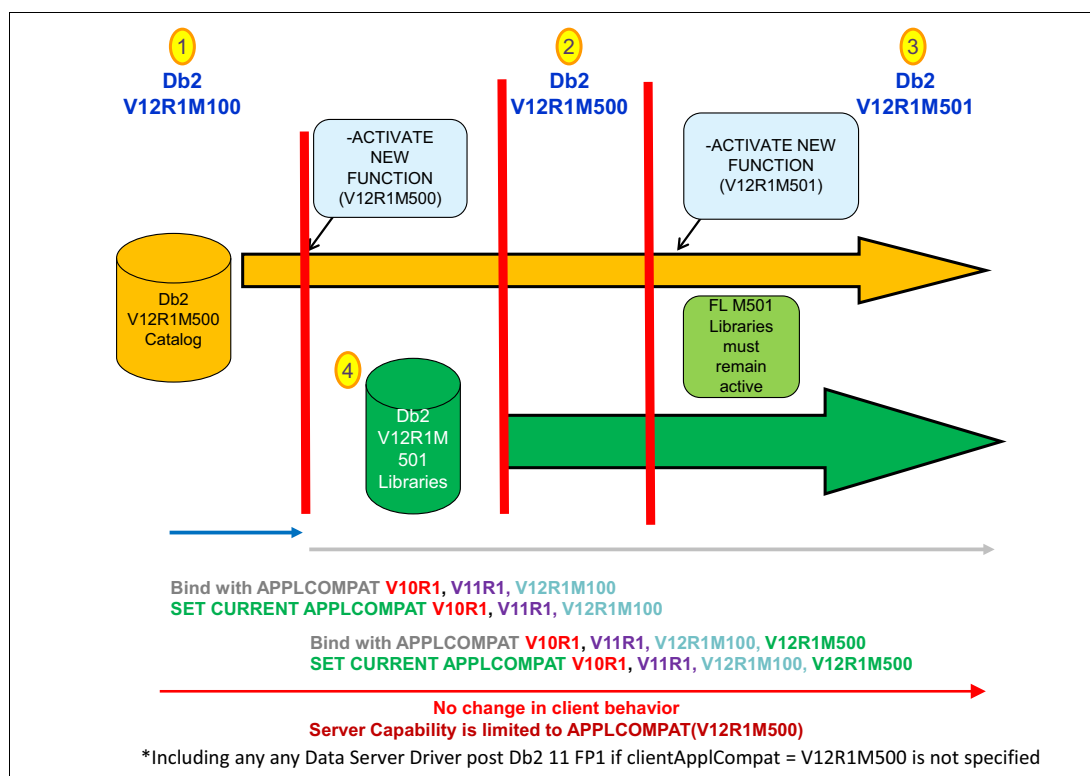


Figure 2-2 Data Server Driver: Pre Db2 11 fix pack 1

In the scenario shown in Figure 2-2, the current version of Db2 is V12R1M100 (Figure 2-2 item 1) is active. The execution of the **-ACTIVATE FUNCTION LEVEL (V12R1M500)** activates Db2 12 modification level M500 (Figure 2-2 item 2). Next, the V12R1M501 libraries are loaded (Figure 2-2 item 4). After the **-ACTIVATE FUNCTION LEVEL (V12R1M501)** command executes the function level is at M501 (Figure 2-2 item 3). The Db2 V12R1M501 libraries must remain active. Because it is not possible to specify `ClientApplCompat` on older drivers, the Data Server Driver packages used by these clients must be bound with `APPLCOMPAT(V12R1M500)` or below. If an attempt is made to specify Data Server Drivers bound with `APPLCOMPAT(V12R1M501)` or above, SQLCODE -30025 is returned.

Figure 2-3 shows the relationship between the `clientAppCompat` keyword and APPLCOMPAT.

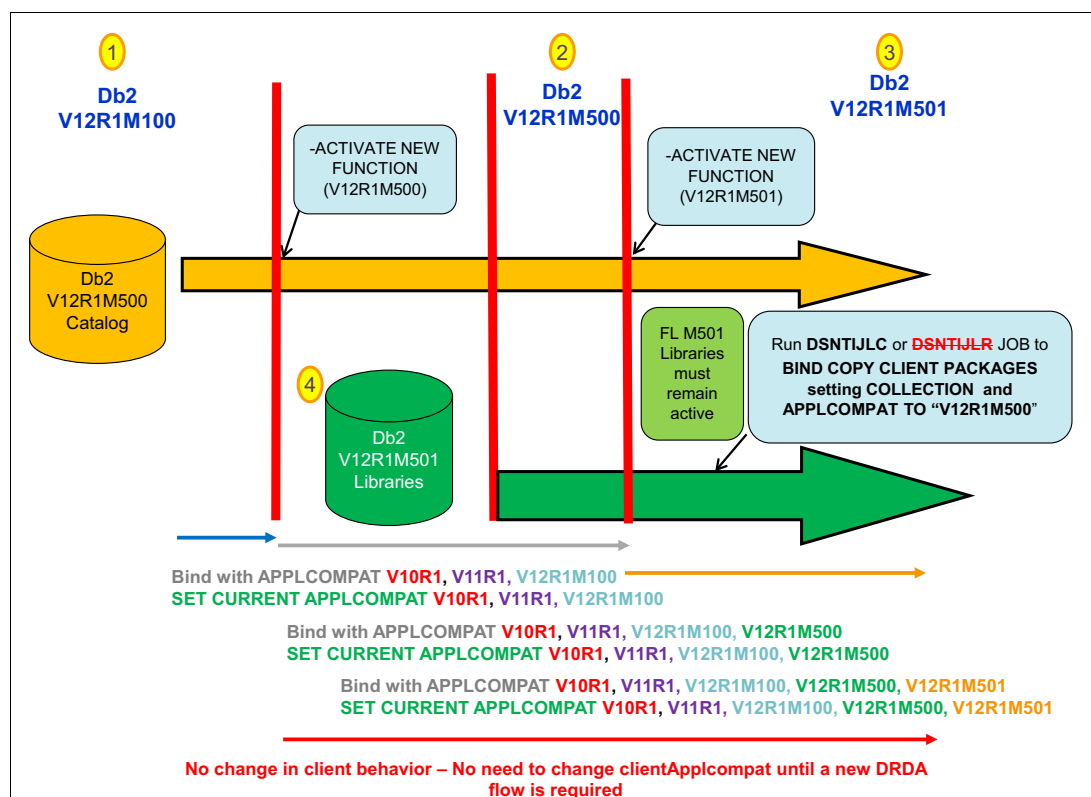


Figure 2-3 Data Server Driver - Db2 11 FP1+

In this scenario (shown in Figure 2-3), the environment was initially running Db2 V12R1M100 (Figure 2-3 item 1). The execution of **-ACTIVATE FUNCTION LEVEL (V12R1M500)** activates Db2 12 function level M500 (Figure 2-3 item 2). Next, the Db2 V12R1M501 libraries (Figure 2-3 item 4) are loaded during normal Db2 maintenance by stopping and starting Db2. The **-ACTIVATE FUNCTION LEVEL (V12R1M501)** command runs updating function level to M501 (Figure 2-3 item 3). The Db2 V12R1M501 libraries must remain active.

It is suggested that DSNTIJLC be run to **BIND COPY** the current Data Server Driver packages into a new COLLECTION. This **BIND COPY** can be executed while existing applications are running and the Data Server Driver packages allocated. Applications can then specify this new COLLECTION, as a property, when they want to use new capability in the recently activated function level. Alternatively, the DSNTIJLR can be run to **REBIND** the Data Server Driver packages with a new APPLCOMPAT level. DSNTIJLR is not recommended, because it can be difficult to rebind the Data Server Drivers without taking an outage.

Important: Additionally, rebinding the Data Server Drivers using DSNTIJLR is not recommended, because it affects all applications connecting to Db2.

Db2 12 only allows the new ClientAppCompat value to be less than or equal to the APPLCOMPAT level of the Data Server Driver packages at the server. At some point in the future, a new server feature will require a new ClientAppCompat level. Let's say a new data type is added in Db2 12 (a new data type is an example of new function that requires a new driver level to use it) in Db2 function level Db2 V12R1M5zz. At that point, client-server applications wanting to use that new capability need to have the following information:

- ▶ Packages bound at the server with APPLCOMPAT(V12R1M5zz).
- ▶ The ClientAppCompat keyword set to V12R1M5zz.
- ▶ A new IBM Data Server Driver client that is capable of supporting the new driver level. In Figure 2-1 on page 33, this client is listed as *Driver Level Vyy FPy*.

Until such a server change that requires a new client to exploit that change, there is no need to change the ClientAppCompat setting to adopt new server side function that can be exploited without a driver change (no need to change the ClientAppCompat setting). Db2 will clearly document any such requirement as part of the function level documentation. You can find information about [application compatibility levels](#) at IBM Knowledge Center.

A ClientAppCompat value that exceeds the server APPLCOMPAT bind option for the Data Server Driver packages results in an error. The error is 30025 EXECUTION FAILED BECAUSE FUNCTION NOT SUPPORTED BY THE SERVER: LOCATION location-name PRODUCT ID product-identifier REASON reason-code (sub-code).

Although you can advance your ClientAppCompat as you increase the APPLCOMPAT of the data server driver packages, it will only be necessary when new client capability needs to be exploited.

Preferred practice: Change your ClientAppCompat only when necessary.

2.3.2 What the clients see after Connect

To ensure that client applications can tell what capabilities of the Db2 server they can use, the following change provides key information:

- ▶ The command line interface (CLI) introduces a new infotype, SQL_DBMS_FUNCTIONLVL, for the SQLGetInfo API that returns the function level or build level.
- ▶ Java Common Client introduces a new getDatabaseFunctionalLevel() API driver that returns a string.

Using these new mechanisms in Db2 products produces the following results:

- ▶ Db2 for Linux, UNIX, and Windows returns the build level (for example "n1604281900").
- ▶ Db2 Z returns the function level (for example "V12R1M504").

2.3.3 SQLCA and GET DIAGNOSTIC impacts

SQL CONNECT is enhanced in Db2 12 to return the Db2 product ID and its current functional level of the connected server in the generated SQLCA. Information about the server is placed in the SQLERRP and SQLERRMC fields.

SQLERRP provides the server product information and, in the case of an error, diagnostic information such as, the name of the module that detected the error. In all cases, the first three characters are “DSN” for Db2 for z/OS. SQLERRP contains the following information:

- ▶ *ppp* is the product identifier and can be one of the following values:
 - AQT: IBM Db2 Analytics Accelerator for z/OS
 - ARI: Db2 Server for VSE and VM
 - DSN: Db2 for z/OS
 - JCC: IBM Data Server Driver for JDBC and SQLJ
 - QSQ: Db2 for i
 - SQL: Db2 for Linux, UNIX, and Windows
- ▶ *vv* is the version identifier, such as 11 for Version 11.
- ▶ *rr* is the release identifier, such as 01.
- ▶ *m* is the modification level.

The SQLERRMC field contains the functional levels. The first token contains the function level of the connected Db2 (for example, *V12R1M506*).

2.3.4 DDF commands and messages

The distributed data facility (DDF) commands and messages that have changed for Db2 12 to show the functional level of each connected system. The commands and messages related to are as follows:

- ▶ DISPLAY LOCATION (shown in Example 2-2)

Example 2-2 DISPLAY LOCATION

```

DSNL200I  -DB2A DISPLAY LOCATION REPORT FOLLOWS-
LOCATION                                     PRDID    T ATT CONNS
SYEC1B                                         DSN12015 S          1
L209-FUNCTIONAL LEVEL=V12R1M500
DISPLAY LOCATION REPORT COMPLETE
  
```

- ▶ DDF Location Statistics
- ▶ DDF Accounting Header



Db2 12 documentation: Information in a new way

With the introduction of continuous delivery in Db2 12, it became clear that without updating the approach to documentation, things might get confusing over the life of a release. Db2 documentation is being updated constantly. Some of the updates are the result of customer feedback, some are clarifications from development, and some changes are the result of new function. With more new functions being added after Db2 12 GA, it was clear that we needed a way to clearly differentiate changes. In the past, change bars were used to indicated changes from a prior release. However, this approach does not scale well when there might be dozens of changes in the documentation several times a year.

The Db2 Information Development team got involved early in the design thinking stages of continuous delivery and produced mock ups to show to customers. Customer feedback was essential and helped to shape the final results. By the time of the Db2 12 GA, we converted the documentation from the previous format for discussing a new release, specifically conversion mode (CM), new-function mode (NFM), and enabling-new-function mode (ENFM), to function levels, with function level 100 allowing coexistence with Db2 11 and function level 500 supporting new Db2 12 capability. With the base infrastructure in place, the Db2 team was ready to document the first function level beyond Db2 12 GA.

The remainder of this chapter discusses the approach used by Db2 to enable documentation of continuous delivery using function level M501 as an example.

Figure 3-1 shows IBM Knowledge Center table of contents for Db2 12 documentation and an overview of function level 501.

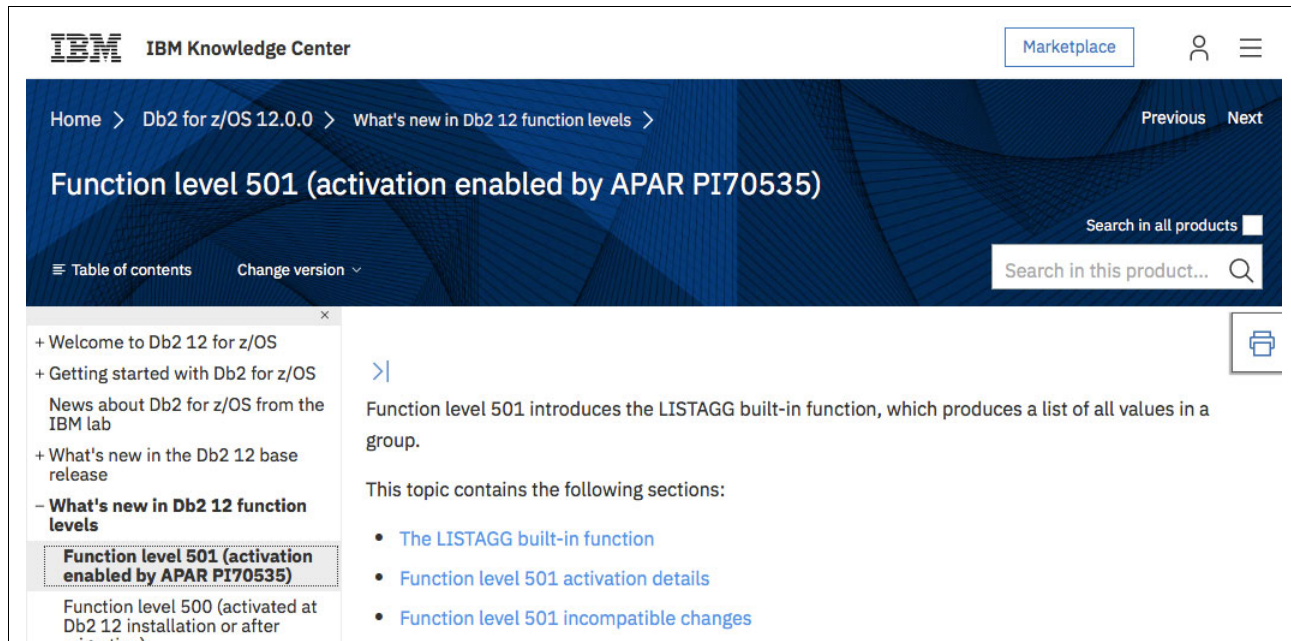


Figure 3-1 Db2 12 Table of contents and function level 501

The information in IBM Knowledge Center includes an introduction to Db2 12 and a description of how enhancements are delivered. There is also a description of function level, starting with the most recent function level at the top of that section. In this case, it is function level 501, which is the most recent function level that is being shipped. Notice there is additional information available, such as the APAR number that activates that function level.

Figure 3-2 shows a little more detail about function levels in particular. Above each function level, it displays a small paragraph about what was delivered in that function level. If you click one of the links, the content expands to the next level of detail, where more information is available about the selected function level.

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 per-requisite or co-requisite PTFs. The activation of a function level results in the activation of all lower function levels.

Although most new capabilities and enhancements in DB2 12 are delivered in function levels. Some continue to be delivered in the base release. For highlighted examples of such enhancements, see [New capabilities highlights DB2 12 base release](#).

The following function levels are available in DB2 12. They are listed in descending order, beginning with the highest available function level:

501

- ➔ **Function level 501 (activation enabled by APAR PI70535)**
Function level 501 introduces the LISTAGG built-in function, which produces a list of all values in a group.
- ➔ **Function level 500 (activated at DB2 12 installation or after migration)**
Function level function level 500 (V12R1M500) represents the first opportunity for applications to take advantage of most new capabilities in the initial DB2 12 release, including new SQL capabilities and subsystem parameter settings. In many respects, function level 500 is analogous to new-function mode in previous releases.
- ➔ **Function level 100 (activated by migration to DB2 12)**
In function level 100 (V12R1M100), DB2 runs on DB2 12 code, and virtual storage and optimization enhancements in the initial DB2 12 release become available. However, most new function, including and subsystem parameter changes and new SQL capabilities remain disabled. In many respects, function level 100 is analogous to conversion mode in previous DB2 releases.

Related concepts:

- ➔ [What's new in the DB2 12 base release](#)

Related tasks:

- ➔ [Adopting new capabilities in DB2 12 continuous delivery](#)

Figure 3-2 A little detail about the function levels

Figure 3-3 shows detail information about the LISTAGG built-in function.

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, YC
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

To learn how to make new capabilities available for use in your DB2® environment, see [Adopting new capabilities in DB2 12 continuous delivery](#).

Figure 3-3 LISTAGG information

In this case, the only item in function level 501 is the new function LISTAGG. Given that LISTAGG is an SQL enhancement, what is shown in Figure 3-3 is a sample query using LISTAGG. For other sorts of capability, for example a command or new Utility syntax, sample syntax and output might be shown. For functions such as a new access path, how that access path works or, for example, EXPLAIN output might be shown. Although not intended to be comprehensive, this information is intended to give a flavor for the new capability and provide links to more detailed documentation.

Figure 3-4 shows more details about function level 501.

Function level 501 activation details	
Table 1. Activation details for function level 501.	
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 V12R1Mnnn application compatibility levels for data server clients and drivers.
Function level 501 incompatible changes Function level 501 introduces no incompatible changes.	
Feedback	

Figure 3-4 More detail about function level 501

In Figure 3-4 there are these important types of information:

- ▶ Enabling APAR: Identifies the enabling APAR in this case P170535 with a link to further details.
- ▶ Information about any required catalog change (if any).
- ▶ Application compatibility Control: Information on application impact, if a new APPLCOMPAT level is needed to exploit the capability, and further information. In this case, a link to the information about Data Server Driver requirements is also included.
- ▶ Function level 501 incompatible changes: This section states information about incompatible changes, if any exist.

Figure 3-5 shows APAR P170535 code to support future new function.

PI70535: CODE TO SUPPORT FUTURE NEW FUNCTION.

A fix is available

Obtain the fix for this APAR.

APAR status

Closed as program error.

Error description

Code to support future new function.

Local fix

Problem summary

* USERS AFFECTED: *
* All DB2 12 for z/OS users. *

* PROBLEM DESCRIPTION: *
* Introducing DB2 12 function level 501. *

* RECOMMENDATION: *
* Apply to all active members to be able to ACTIVATE function *
* level 501. *

Enables activation of DB2 12 for z/OS function level 501.

For detailed descriptions of the new DB2 12 capabilities in
function level 501, and specific activation details, see
"Function level 501" in IBM Knowledge Center:
<http://ibm.biz/db2z12functionlevels>

For an introduction to DB2 12 continuous delivery concepts and
instructions for adopting new capabilities in DB2 12, see:
[http://www.ibm.com/db2z12continuousdelivery](#)


<http://www-01.ibm.com/support/docview.wss?uid=swg1P170535>

Page 1 of 3

Figure 3-5 Information about P170535

The APAR itself has just basic information about the function level and a link to IBM Knowledge Center with the details. This is different than how enhancements were typically described in APAR closing text in the past. In the past closing text for new function APARs (or APARs that enabled new function that already has shipped in other APARs) was verbose. It was common to see ++HOLD and ++DOC information describing actions or information that were needed to use the new capability. With the introduction of function levels, the APAR no longer needs that level of documentation, because IBM Knowledge Center has extensive documentation regarding activities about activation of a function level. Although it might seem like a ++HOLD would (or should) be included, the act of applying the APAR enables the system to be able to -ACTIVATE the function level. Applying the APAR has no effect on the system; it is the activation of that function level, which typically occurs at a much later date, that effects the system.

Figure 3-6 shows information about application compatibility levels for data server clients and drivers for a specific Db2 12 function level.



V12R1M nnn application compatibility levels for data server clients and drivers

Version 12.0.0 ▾

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

Data server clients and drivers must be at the following levels to exploit DB2 for z/OS *function-level* application compatibility of greater than V12R1M500:

- IBM® Data Server Driver for JDBC and SQLJ: Versions 3.72 and 4.22, or later
- Other IBM data server clients and drivers: DB2 for Linux, UNIX, and Windows Version 11.1 Modification 1 Fix Pack, or later

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. For clients packages that contain static SQL statements, rebind the static packages with the new application compatibility value, on the client and on the server.
2. For client and driver packages that contain only dynamic SQL statements, bind or rebind the

https://www.ibm.com/support/knowledgecenter/en/SSEPEK_12.0.0/apsg/src/tpc/db2z_applcompatclients.html?view=embedPage 1 of 3

Figure 3-6 New Data Server Drivers

This section of the documentation (Figure 3-6) is an extension of the application impact information and is specific to the IBM Data Server Drivers. This information is specific to function level 501 activation and, likely, would not be present for all function levels.



Final thoughts on continuous delivery

The infrastructure that the Db2 development team put in place for Db2 12 enables customers to receive preventive service, prescriptive service, and new function in the same stream. The new function level concept gives customers the choice of when to enable access to new function. In general, customers adopt a new function level after they are satisfied that their maintenance level is stable.

The APPLCOMPAT bind option provides an island of stability for an application. Customers can develop, test, and deploy an application without exposing that application to changes across releases and function levels. The islands of stability that APPLCOMPAT provides an application are an important capability that will enable the stability of applications that Db2 customers expect in their production systems.

This paper includes the following key points:

- ▶ Db2 12 enables the continuous delivery of new function in the service stream, alongside of preventive and prescriptive maintenance.
- ▶ Enabling new function is controlled by use of:
 - The **CATMAINT** utility
 - The **-ACTIVATE** command
 - The APPLCOMPAT BIND option
- ▶ APPLCOMPAT is extended in Db2 12 to apply to DDL and DCL.
- ▶ The IBM Data Server Driver Packages (such as, ODBC, JDBC, .Net, PHP, RUBY, and SAP) can be controlled with APPLCOMPAT and the `clientApplCompat` configuration keyword.
- ▶ Customers should use collections and DSNTIJLC to ensure that applications using the IBM Data Server Drivers remain stable for all function levels.
- ▶ The SYSIBM.SYSLEVELUPDATES table contains history regarding when system changes occurred.
- ▶ Documentation is changing to help highlight new functions.
- ▶ The Db2 development team is working with ISVs to improve communication, to provide early notification of changes, and to give access to PTFs prior to general availability.

With the capabilities in Db2 12, customers can now safely adopt changes at a pace of their choosing that Db2 is shipping in the maintenance stream. Db2 maintenance will contain enhancements, alongside preventive and prescriptive changes. Customers will not be required to adopt a function level to apply maintenance. Enterprises can now embrace their DevOps for applications adopting the latest Db2 capabilities while still maintaining stability for existing applications.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this paper.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only.

- ▶ *IBM Db2 12 for z/OS Performance Topics*, SG24-8404
- ▶ *IBM DB2 12 for z/OS Technical Overview*, SG24-8383
- ▶ *DB2 12 for z/OS Technical Overview and Highlights*, REDP-5444
- ▶ *DB2 12 for z Optimizer*, REDP-5445

You can search for, view, download or order these documents and other Redbooks, Redpapers, web docs, draft and additional materials, at the following website:

ibm.com/redbooks

Online resources

These websites are also relevant as further information sources:

- ▶ IBM Knowledge Center, Db2 12 for z/OS documentation
https://www.ibm.com/support/knowledgecenter/SSEPEK_12.0.0/home/src/tpc/db2z_12_prodhome.html
- ▶ IBM Support, Db2 for z/OS product information
<https://www.ibm.com/support/docview.wss?uid=swg27047206>
- ▶ Db2 12 for z/OS product information
<https://www.ibm.com/analytics/us/en/db2/db2-for-zos/>

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services



REDP-5469-00

ISBN 0738456713

Printed in U.S.A.

Get connected

