IBM Db2 12 for z/OS Function Level Activation and Management

Gareth Copplestone-Jones

Triton Consulting

June 2021

Executive Summary

This executive summary provides a brief introduction to Continuous Delivery in Db2 12 for z/OS and future releases and addresses some of the key issues faced by decision makers who are responsible for project approval and funding, or who have management responsibility for the delivery of the production service.

Most if not all Db2 for z/OS releases have introduced new function in the maintenance stream throughout the lifecycle of the release. Db2 12 for z/OS is no exception. Where Db2 12 differs from other releases in this regard is with the introduction of Continuous Delivery. This allows IBM to deliver new functionality more quickly and allows their customers to exploit that new functionality in a controlled way. The four components that make up the Continuous Delivery mechanism are presented in the side-bar.

Anecdotal evidence that a significant number of Db2 sites were slow in advancing function levels led the IBM Gold Consultants to launch a survey to gain a better insight into the different continuous delivery strategies being employed. Although the sample size was small, it became clear that continuous delivery in Db2 12 for z/OS and the mechanisms for delivering it raise a number of questions and concerns for decision makers.

- It is vitally important that continuous delivery does not present a risk to the production service in terms of resilience and degraded performance.
- The behaviour and performance of existing applications must not change when maintenance is applied, when changes are made to the catalog or when the function level is advanced.

The Maintenance Level. This provides some indication of what maintenance has been applied to Db2. It also indicates the Function Levels and Catalog Levels that the system is capable of supporting. Defect fixes and new function are integrated in a single maintenance stream, and IBM have guaranteed that defect resolution is not dependent on function levels.

The Catalog Level. This indicates what changes that have been made to the Db2 catalog. It is dependent on the Maintenance Level, although not all Maintenance Levels have an equivalent Catalog Level. Function Levels have an associated minimum Catalog Level.

Function Level. This indicates the functional capability of the standalone Db2 subsystem or Db2 data sharing group. The Function Level is dependent on the Catalog Level and the Maintenance Level.

The Application Compatibility Level (APPLCOMPAT). This is a BIND option and indicates the functional capability of the application package in terms of SQL syntax and behaviour. It is dependent on the Function Level.

It is also clear from discussions with IT executives and leaders in addition to the survey findings that the financing and resourcing of the project management, implementation and testing required for continuous delivery projects are a major factor in determining the continuous delivery strategy. For many businesses, not only are infrastructure changes competing with application changes for change slots, but they are competing for funding in a culture where application projects are seen as delivering far more business value than infrastructure projects.

It is clear both from the survey results and from the discussions that concerns about planning and scheduling change, restricting the impact of continuous delivery on the system and the applications, and budgetary constraints are the main reasons why some businesses are reluctant to advance function levels. These concerns cannot be simply dismissed out of hand, but to some degree they arise out of some widely held misunderstandings about the impact of continuous delivery. This

summary therefore provides a series of statements about continuous delivery and in particular about advancing function levels that decision makers will find useful when planning or reviewing their Db2 for z/OS continuous delivery strategy.

Staying at the lowest possible function level (V12R1M100) does not freeze all new functions and features. Defect fixes and new function are integrated in a single maintenance stream, and the majority of Db2 12 for z/OS enhancements such as performance improvements and additional utility capabilities are delivered in the maintenance stream and are available at all function levels. This not unusual, as new features and functions have been delivered in the maintenance stream in all previous Db2 releases, at least since Db2 V4.1.

Advancing function levels is required to be able to migrate to the next version of Db2 for z/OS. As stated by IBM when Db2 12 for z/OS became generally available, continuous delivery did not mark the end of new Db2 for z/OS versions. For businesses who want or need to be on current product release levels, advancing function levels is mandatory to be able to migrate to the next release of Db2 which is expected to become generally available in 2022. At some point after this, support for Db2 12 for z/OS will be withdrawn.

Nearly all function-level controlled new non-SQL features can be switched off so that, even though the function level is activated, the new features and functions are inactive and can be selectively turned on at a time and with a scope of your choosing. For that small number of incompatible or unavoidable changes introduced by some function levels, this document outlines the specific areas where these apply and the mitigating actions to take to minimise or eliminate their impact.

APPLCOMPAT controls which applications are exposed to new SQL features, functions, and behaviour, not the Function Level. It isolates existing applications from being exposed to new SQL features or changes in behaviour.

It is not mandatory to advance APPLCOMPAT for existing applications. The only reason to advance APPLCOMPAT is if you modify the application to exploit the new SQL syntax or new SQL behaviour. In fact, it is recommended that you do not rebind existing application packages with a higher APPLCOMPAT as there is no benefit in doing so.

Dynamic SQL packages can support multiple APPLCOMPAT settings by keeping a separate copy for each APPLCOMPAT setting. This can be done without modifying the application.

Activating function levels does not require the same level of testing as release migration. Because you can turn off nearly all new non-SQL features, and because APPLCOMPAT controls what SQL features are available at the application program level, the testing requirements for function-level activation are more like those for a maintenance roll-out. In some cases, where you are planning to exploit new function, you may need additional testing, but if you don't currently perform full application regression testing for a maintenance roll-out, then there is no need to do so for function level activation.

Activating a function level is not a disruptive change. While running the catalog maintenance process in preparation for activating a function level can be disruptive because of catalog contention, activating a function level is not disruptive change and does not require a Db2 restart. The recommended best practices for continuous delivery section of this document provides guidance in minimising continuous delivery disruption.

More detailed information about the above topics, including recommended continuous delivery practices, can be found in the rest of this technical paper.

Table of Contents

Tables

Figures

Introduction

Db2 12 for z/OS became generally available on October the 21st, 2016. The new version included a radical change in the way new features and functions are delivered, Continuous Delivery. Continuous Delivery delivers new features and functions in the maintenance stream throughout the lifecycle of a Db2 release and provides users with the capability to control the activation of new features at the system and application levels.

Anecdotal evidence suggests that, for some businesses, activating new function levels is part of their standard procedures, whereas other businesses remain at the lowest possible function level, V12R1M100. In some cases, this seems to be based on a misunderstanding that Continuous Delivery indicated that Db2 12 for z/OS would be the last new version, with all new features and functions delivered in the Db2 maintenance stream. However, this was never IBM's intention – Db2 12 for z/OS has been generally available for nearly 5 years, and a new version, codenamed VNext, is in development. This is at least in part to provide the architectural support for a major new theme, AI and Hybrid Cloud exploitation. To be able to migrate a Db2 for z/OS subsystem or data sharing group to the new version, Db2 12 for z/OS function level V12R1M510 must have been activated. ***Implementing a strategy for function level advancement has therefore become much more important***, as businesses planning to migrate to the new Db2 version must be prepared to advance function levels prior to migration.

In April 2021, the IBM Gold Consultants launched a survey to try to understand the how widespread the adoption of new function levels is among the Db2 user community, given that there is some anecdotal evidence that many have adopted a conservative strategy of remaining at function level V12R1M100. We wanted, therefore, to get a clearer picture of what function levels have been activated, what the plans for future function level activation are, and what the user community perceives as being the biggest challenges when devising a function level activation strategy.

The Survey Sample

The sample size used for the survey was limited. This means that the survey results cannot in any way be regarded as representative of the entire Db2 user base. The objective of the survey was to gain an insight into current Db2 12 for z/OS practices.

This Paper

This paper is, in part, the output of that survey. It is targeted at Db2 for z/OS technical professionals, IT managers, executives, and project managers. Its objectives are four-fold:

1. To gain an insight into the function level activation strategy adopted by businesses, including the key reasons some have adopted a strategy of remaining at function level V12R1M100.
2. To address the concerns about advancing function levels.
3. To provide guidance on advancing functions safely while minimising the impact on the production service.
4. Ultimately, to promote the proactive activation of function levels.

It therefore discusses the following topics:

- A high-level view of the issues around function level advancement:
    - The key survey results.
    - The underlying reasons why some businesses are reluctant to advance function levels.
    - Function level-independent new features.

- A series of more technical topics:
    - Function level-controlled SQL-related features and application compatibility.
    - Updates to the Db2 catalog – CATMAINT.
    - Db2 12 function-level dependent non-SQL Features and how to manage them.
    - Incompatible/unavoidable Db2 12 for z/OS function level changes and how to manage them.
- Information on Continuous Delivery recommended practices, including:
    - Testing
    - Advancing from FL100 to FL500 or FL501
    - Advancing beyond FL500/FL501, including CATMAINT
    - Managing application compatibility, including generic dynamic SQL packages with special consideration for DRDA packages.
- An appendix on the Continuous Delivery mechanism, including maintenance levels, catalog levels, function levels and application compatibility.
- An appendix with the survey results.

While every effort has been made to validate all the information in this document, its accuracy cannot be guaranteed. This document is therefore subject to correction and amendment as need arises.

Acknowledgements

This paper is the outcome of a series of discussions between a group of IBM Gold Consultants with an interest in promoting the adoption of function levels by Db2 for z/OS users. It was mostly written by Gareth Copplestone-Jones of Triton Consulting, but could have not been completed without the help of the following people, who engaged in the discussions, gave the benefit of their experiences with Db2 12 continuous delivery, reviewed and corrected the paper, and contributed to the survey, both in terms of setting it up and providing questions, and in terms of analysing the results:

- **Julian Stuhler** of Triton Consulting
- **Adrian Collett** of Expertise4IT S.r.l**.**
- **Dave Beulke** of Dave Beulke and Associates
- **Frank Filmore** of The Filmore Group

The author and contributors would like to than the other IBM Gold Consultants who participated in additional reviews and discussions.

A Note on Terminology

A variety of different terms and notational forms are used when talking about maintenance levels, catalog levels, function levels and application compatibility. The IBM documentation refers to both ‘code levels’ and ‘maintenance levels’, which are the same thing: an indicator of the fixes that have been applied to Db2 for z/OS; and an indicator of which function levels can be activated on the system. To avoid confusion, this document always uses the term ‘maintenance level’. When referring to the various levels themselves, the full notation for Db2 12 for z/OS is in the form V12R1Mnnn where ‘nnn’ is a represents the level. For example, V12R1M100 can be used to refer to Db2 12 for z/OS function level 100. For short, it’s variously referred to as function level M100, function level 100, and FL100. This document uses the FLnnn notation as well as the full notation for function levels, but always uses the full notation to refer to maintenance level, catalog level and application compatibility.

Key Survey Results

While the results of the survey might not be statistically significant, they nevertheless provide a valuable insight into the management of Db2 12. This section discusses those survey results which are, in the opinion of the authors, the most important:

- The understanding of Continuous Delivery and its four pillars, the maintenance level, the catalog level, the function level, and the application compatibility or APPLCOMPAT level.
- The adoption of function levels in the user community.
- The policy agreed within the business for advancing function levels.
- The challenges inhibiting function level activation in the user community.

The Understanding of Continuous Delivery

Db2 12 for z/OS became generally available (GA) on October the 21st 2016. Despite that the survey results indicate that there is still some confusion about Continuous Delivery and its four pillars, the maintenance level, the catalog level, the function level, and the APPLCOMPAT level.

This is reflected in the reporting of maintenance levels, without around 15% of respondents reporting a maintenance level that is nearly 5 years old – the maintenance level when Db2 12 was GA, as demonstrated in Figure 1 Maintenance Levels.

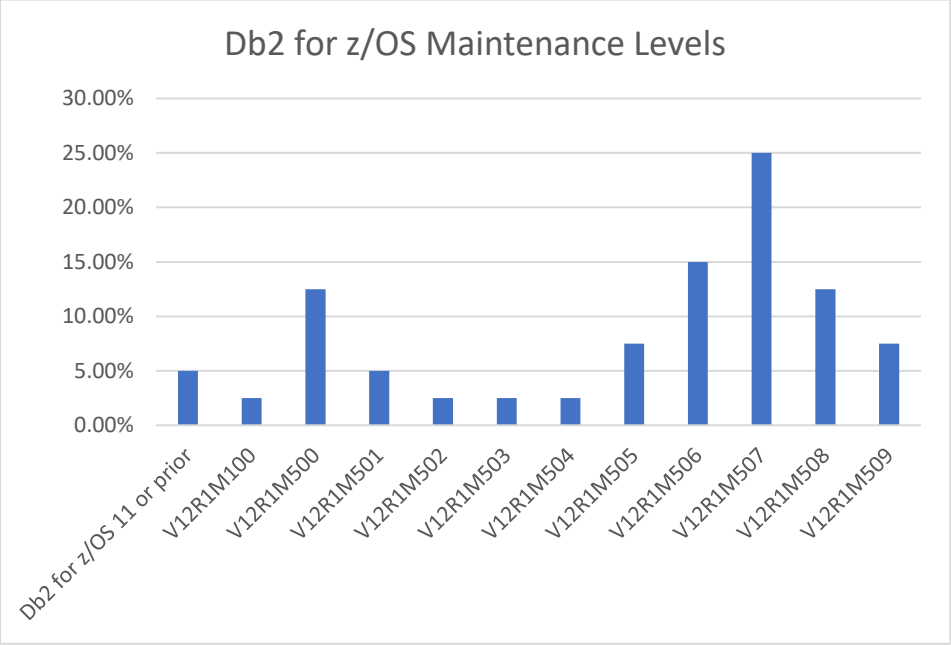

Figure 1 Maintenance Levels

If the reported maintenance levels are correct, it first of all means that about a third of Db2 sites are on a maintenance level that is at least two years old – the maintenance required for FL504 activation came out in April 2018. This alone should be cause for concern, as these sites are ***missing a number of important fixes.*** However, it also appears that there is some confusion about the maintenance level, as 2% of respondents reported they are on a non-existent maintenance level, V12R1M100: the maintenance level of Db2 12 for z/OS on the GA date was actually V12R1M500. A new Db2 12 for z/OS system is installed at FL500, which requires a minimum maintenance level of V12R1M500.

There also seems to be some confusion about catalog levels, with non-existent catalog levels being reported, and required ones omitted. See Figure 6 Db2 for z/OS Catalog Levels in the Appendix.

The Adoption of Function Levels in the User Community

While there are still some businesses running on Db2 11, they do appear to have plans to migrate to Db2 12. Of those who are on Db2 12, over 20% are still on function level FL100. A similar number are on function level FL500, with an additional 10% having dipped their toe into Continuous Delivery with virtually no-risk FL501. That accounts for over 50% of the user community, with only 40% having advanced to FL502 or higher.

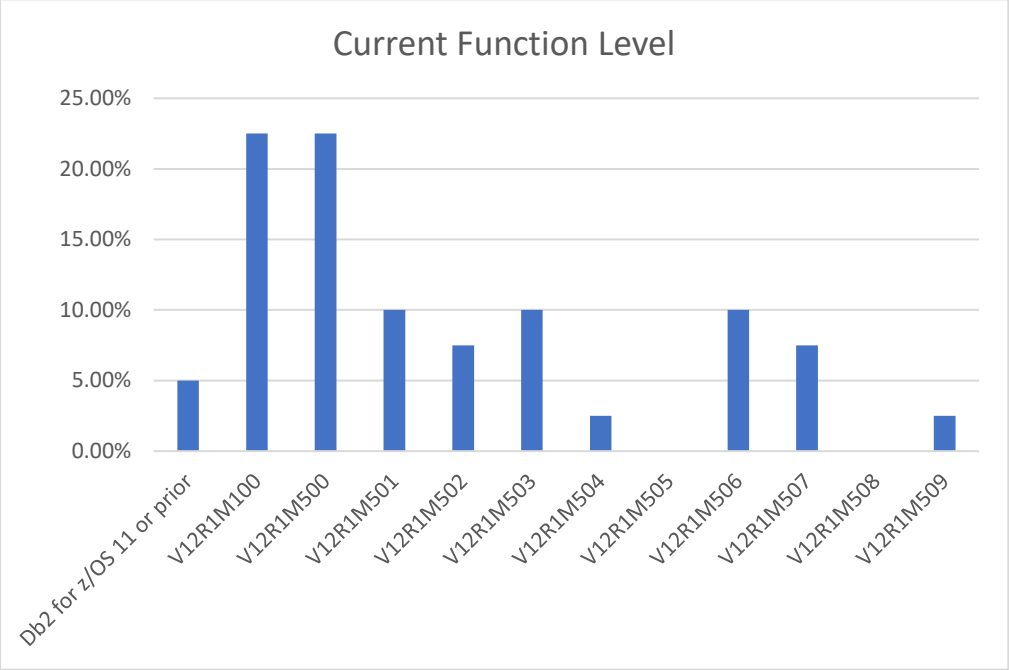

Figure 2 Current Function Level

More encouraging are the expectations of the user community that the vast majority will have activated FL500 or higher in the next six months, with a significant number planning to be at FL509. We are being cautious in interpreting these responses, as it's possible that at some of these expectations will not be met. See Figure 8 Planned Function Level in the Appendix.

The Agreed Policy for Advancing Function Levels

The survey asked respondents what their business's agreed policy for advancing function levels is. Of those who replied, nearly ***40% stated that they have no agreed policy***, which raises a number of questions, such as why just over nearly 75% have advanced to FL500 or higher, which suggests that a significant proportion of businesses ***have advanced function levels without having an agreed policy on when and how to do so***. Of the remaining approximately 60% of the respondents, just under 35% said that the policy was based on the maintenance schedule, while only just over 25% said the policy was driven by the demand for specific features.

Challenges to Function Level Advancement

The survey asked respondents to list the two biggest challenges to advancing levels faced by their business. We then divided those responses into categories, to try to get a clearer understanding of what is preventing customers from advancing function levels:

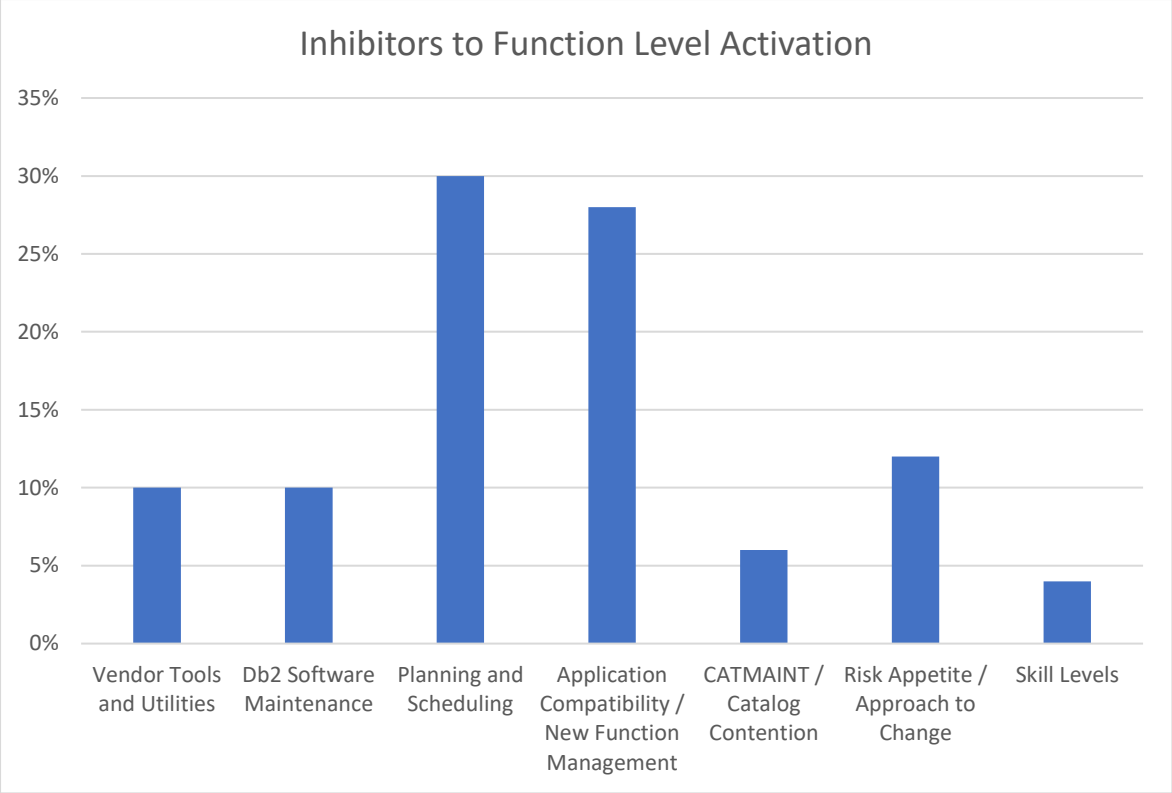

Figure 3 Inhibitors to Function Level Activation

The following example comments help to illustrate how the challenges are categorised, as well as providing some useful insights. Some of the comments have been edited to improve readability, but their essential meaning has not been changed.

Example comments for “Vendor Tools and Utilities:

- “Making sure all products are compatible”
- “All tools have to be on the right level before activating a function level”

Example comments for “Db2 Software Maintenance”:

- “Hitting PEs”
- “I applied maintenance just prior to wanting to advance to some new function and ran into a show stopping bug. Therefore, I can't put that on production (M507). I want to get more current before going past M100.”
- “The usage of the new functions because they are not always stable”

Example comments for “Planning and Scheduling”:

- “Length of time end to end - to apply required maintenance, and activate and verify through all Dev, Test, PreProduction and Prod Subsystems”
- “Application managers require 1 year lead time to budget contractors to test new features”
- “Requires an outage and we don't see any advantage to increasing the function level”

Other comments for “Planning and Scheduling” referred to competition with application and other infrastructure implementations for change slots and testing slots.

Example comments for “Application Compatibility / New Function Management”:

- "Coming up with a plan to manage APPLCOMPAT levels - application and NULLID"
- "Making sure the application teams upgrade their applications and make … [them] compatible with the newer function levels"
- "Understanding SQLLEVEL, and how DDF work needs to be handled"

Example comments for "CATMAINT / Catalog Contention":

- "Our biggest overall challenge is contention with the catalog when trying to perform the CATMAINT. There are no days of the week or times of day where the catalog is free from contention. We need a solution for this that will allow our current processes to continue uninterrupted."
- "Fallback below current catalog level not possible"

Example comments for "Risk Appetite / Approach to Change":

- "We have very high SLAs with our customer. We move very slowly when it comes to upgrades, to make sure we do not impact production processing."
- "The application teams are not exploiting anything new in DB2 for z/OS. We look to them for new functionality they may need."
- "Fear of change"

While categorising the responses has helped provide insight into the challenges faced by businesses, it is possible to draw out some common themes. There is a relatively widely held view among departmental and executive level IT managers and, to a lesser extent, Db2 technicians, that ***advancing function levels is equivalent to a mini-release roll-out***, or even to migration ***to a new Db2 release***. The concerns are:

- Exposing the system and applications to significant new function introduces risk when availability of the production service is all important.
- Advancing function levels is perceived as a major project requiring the following:
    - Significant effort by the Db2 systems programming/DBA team.
    - Updates to the Db2 catalog which could cause contention and disrupt the production service.
    - Additional change slots for CATMAINT and for advancing the function level.
    - Full regression test of the entire set of applications prior to rolling out in production.
- There are a number of incompatible changes and changes that can't be avoided when advancing function levels.
- Although the fact that APPLCOMPAT isolates application migration from system migration, ***some businesses still take the view that applications are affected by function level activation***, that APPLCOMPAT levels need to be advanced, with all the additional testing requirements that implies.

While a number of incompatible or unavoidable changes are introduced when you advance function levels, there are four good reasons why doing ***this is more like a maintenance roll-out*** than a mini-release roll-out:

- Many of the functions and features only available through advancing function levels are SQL-related. In order to exploit those SQL features, you have to bind the packages for your application programs with the appropriate application compatibility (APPLCOMPAT) level. Limiting setting higher APPLCOMPAT levels to only those applications that need to exploit the new features avoids the need for a full regression test of all applications.

- ***RE***Binding (not BINDing) static SQL packages with a higher APPLCOMPAT level brings ***NO benefit*** whatsoever – any new function is only available after program modification and therefore a BIND, and any optimizer enhancements are introduced with the maintenance level. If you are resolving incompatibilities or allowing for changes behaviour, you will need to modify the application and use BIND with REPLACE.
- In any case, much of the new function delivered in the maintenance stream is not dependent on a function level and is available immediately after a maintenance roll-out.
- The CATMAINTs required to advance to a catalog level high enough to support a new, higher function level are less disruptive and less likely to lead to contention than the CATMAINT you run to migrate to Db2 12.
- Those function level-delivered new features and functions that aren't related to SQL can be controlled via the Db2 system parameters (ZPARM) or other means.

Handling the incompatible or unavoidable changes is covered in Incompatible/Unavoidable Db2 12 Function Level Changes, which examines where they apply and discusses what actions you can take to mitigate them.

Approaches to testing are covered in the section on Function Level Activation Recommended Practices.

Function Level-independent New Features

As with previous Db2 releases, a significant amount of new function is delivered in the maintenance stream. Some of that new function is only available when you advance to the relevant function level, as with the support for encryption and decryption built-in functions using key labels. However, IBM themselves have stated that 80% of the new features and functions delivered in the maintenance stream are available at all function levels, including V12R1M100. Therefore***, not advancing function levels does not isolate you from all new functions and features***.

Function Level-controlled SQL-related Features And APPLCOMPAT

Db2 11 introduced application compatibility, or APPLCOMPAT, which was intended to separate system migration from application migration when upgrading to a new release. For example, once you had upgraded to Db2 11 for z/OS by migrating the system, you could choose to delay application migration and ensure that the applications continued to behave as they did under Db2 10. This had two aspects:

- You didn't need to resolve any incompatibilities related to SQL DML when migrating to Db2 11. You could make any necessary changes according to your own schedule, and then bind the packages for those application programs with APPLCOMPAT(V11R1). In fact, there was no immediate need to resolve those inconsistencies, as Db2 12 also supports an APPLCOMPAT level of V10R1.
- No new SQL DML capabilities were available until you resolved any inconsistencies and could bind the exploiting packages with APPLCOMPAT(V11R1).

Db2 12 takes that a stage further, firstly by introducing support for function levels as well as releases, and extending application compatibility to control SQL DDL and DCL as well as DML. Even when you advance function levels, new SQL features are not available until you modify the application and bind the relevant packages with the associated APPLCOMPAT setting. For example, to exploit the new built-in SQL DML functions for encryption and decryption with key labels, you not only have to advance to function level FL505, but you also have to update the affected application

programs to use the new built-in functions and bind the associated packages with an APPLCOMPAT attribute of V12R1M505.

Updates to the Db2 Catalog – CATMAINT

When you run CATMAINT as part of the process to migrate to Db2 12 FL100, the catalog level is V12R1M500. Some of the function levels from FL502 onwards also need CATMAINTs. None of those CATMAINTs introduce incompatible changes, and none of the changes involve any significant restructuring of the catalog. The IBM Knowledge Centre lists all the Db2 12 catalog changes, but to summarise, the changes by catalog level are as follows:

- **V12R1M502** – column KEYLABEL added to four tables
- **V12R1M503** – new global variable added
- **V12R1M505** – previously unused column COPYID now used in three tables
- **V12R1M507** – new global variable added
- **V12R1M509** – changed column COMPRESS in two tables and new column COMPRESS_USED added to one table.

Compare that to the many times larger number of changes for FL100, catalog level V12R1M500, with well in excess of 200 individual changes (new tables, new and changed columns, new indexes, etc).

While the possibility of CATMAINT disrupting the production service because of contention between CATMAINT and the applications remains a concern for many organisations, ***APAR PH28472*** should offer some relief. It was made available in September 2020 and is intended ***to increase concurrency between CATMAINT and application processes*** by reordering the CATMAINT SQL statements to reduce the chance of deadlock.

Bear in mind that before you can run CATMAINT in preparation for advancing to function levels FL502 or higher, you must first activate function level FL500 or FL501. Activating FL500 or FL501 does not require a CATMAINT.

New Function Not Managed By APPLCOMPAT

With the exception of FL500, the majority of new features are related to SQL and only available when the application is bound with an APPLCOMPAT equal to or higher than the function level that introduced the new feature. However, there are a number of features not controlled by APPLCOMPAT. Many of those can be switched off using system parameters (ZPARMs) or other controls, and these are covered in the section on Db2 12 Function-level Dependent Non-SQL Features. Of the new features not managed by APPLCOMPAT, several are either incompatible or unavoidable changes that apply in limited circumstances. These are discussed in more detail in the section on Incompatible/Unavoidable Db2 12 for z/OS Function Level Changes, including any mitigating steps you might need to take.

Check the Compatibility of Your Vendor Tools

If you use vendor tooling for Db2 housekeeping, schema change or replication, check with your vendors for the compatibility of their tools with the Db2 12 function levels.

- IBM provides a comprehensive matrix of the compatibility of their tools with Db2 12 function levels.

- Broadcom provide a comprehensive matrix of the compatibility of all their products with Db2 releases, with links to additional information. They also provide information about CA Database Management Solutions for DB2 for z/OS.
- BMC and other vendors – too many to list here – provide similar information.

Most vendor products provide backwards compatibility, so you should be able to perform any necessary tool upgrades prior to and independently of advancing function levels. In all cases, contact your vendor directly to make sure you have the latest information.

Db2 12 Function-level Dependent Non-SQL Features

The following table provides recommendations, where possible, for preventing or controlling the use of function-level dependent new features and functions which are independent of the package APPLCOMPAT attribute. Recommendations to use the same setting for all members are not enforced by Db2. There are some incompatible changes that require resolving for affected applications, and some unavoidable changes which come into effect when you activate the function level. These are highlighted in red and are discussed in more detail in the section on Incompatible/Unavoidable Db2 12 for z/OS Function Level Changes immediately after the table. The list of features excludes migration process changes, Db2 AI for z/OS support, and new or changed console messages.

Table 1 Function-Level Controlled Incompatible Changes

Function Level	New Feature	Comments
V12R1M500	Fast Traverse Blocks (FTBs). INDEX_MEMORY_CONTROL ZPARM	FTBs become effective for GBP dependent indexes at V12R1M500. To avoid their use when activating V12R1M500, specify ZPARM INDEX_MEMORY_CONTROL=(SELECTED,AUTO) and ensure there are no indexes with column ACTION set to 'A' in the SYSIBM.SYSINDEXCONTROL catalog table. It is recommended that you use the same setting for all members.
	AUTH_COMPATIBILITY ZPARM	This specifies whether the new UNLOAD privilege is used for authorisation checking when executing the UNLOAD utility. Do not use the default setting (blank) for pre-V12R1M500 behaviour – instead, specify SELECT_FOR_UNLOAD which causes the UNLOAD utility to check whether the user has the SELECT *or* the UNLOAD privilege on the target table.
	CACHEDYN_STABILIZATION ZPARM	Use a setting of 'NONE' to prevent accidental use. Other settings are only effective when statements are captured following a START DYNQUERYCAPTURE command. It is recommended that you use the same setting for all members.
	COMPRESS_DIRLOB ZPARM	Use the default setting of NO to avoid compressing the Db2 directory LOBs. It is recommended that you use the same setting for all members.
	DDL_MATERIALIZATION ZPARM	Use the default setting of 'ALWAYS_IMMEDIATE' for existing behaviour. It is recommended that you use the same setting for all members.
	Faster insert for concurrent	Use '0' to disallow use of Db2 12 insert algorithm 2,

Function Level	New Feature	Comments
	insert-intensive applications. DEFAULT_INSERT_ALGORITHM ZPARM.	use '1' (the default) to allow selective use at the tablespace level.
	Relative page numbering for UTS PBR tablespaces. PAGESET_PAGENUM ZPARM.	Specifies the default page numbering scheme for UTS PBR tablespaces where not specified in CREATE TABLESPACE. Use the default setting of 'ABSOLUTE' for pre-V12R1M500 behaviour.
	Block specific Db2 Utilities for replication-enabled tables. UTILS_BLOCK_FOR_CDC ZPARM.	Use the default setting of 'NO' for pre-V12R1M500 behaviour.
V12R1M502	z/OS DFSMS data set encryption using key labels.	Requires action outside Db2 to turn on data set encryption. Avoid using DFSMS to encrypt data sets using key label management.
V12R1M503	Temporal auditing	**This is an incompatible change.** It applies to system-period temporal tables defined with ON DELETE ADD EXTRA ROW. See the IBM Documentation website for more information on Function level 503 and Scenario for tracking auditing information .
	The Db2-supplied stored procedure for system time overrides is no longer supported.	**This is an incompatible change.** The SET_MAINT_MODE_RECORD_NO_TEMPORALHISTORY stored procedure which stops the writing of history rows and allows for the override of GENERATED ALWAYS columns is no longer supported. **This stored procedure is still supported if you run with APPLCOMPAT V12R1M502 or lower.**
V12R1M504	Huffman compression. TS_COMPRESSION_TYPE ZPARM	Use the default setting of 'FIXED_LENGTH' to continue using Db2's pre-V12R1M504 compression algorithm. Although not specified in the IBM documentation, I recommend that you use the same setting for all members.
V12R1M505	Reducing latency between Db2 for z/OS and IDAA	**Unavoidable change** for improving performance in Hybrid Transactional Analytical Processing (HTAP) environments.
	Package REBIND phase-in	**Unavoidable change** that allows Db2 to rebind a package concurrently with execution of the package.
	Improved RUNSTATS performance, changed STATPGSAMP ZPARM.	**This is an incompatible change.** The default value for STATPGSAMP, 'SYSTEM' changes from meaning 'NO' prior to V12R1M505 to meaning 'YES'. Change this to 'NO' to continue with the pre-V12R1M505 behaviour. Although not specified in the IBM documentation, it is recommended that you use the same setting for all members.
V12R1M506	Implicit dropping of explicitly created tablespaces (UTS and LOB)	**Unavoidable change.** If you have ZPARM REORG_DROP_PBG_PARTS set to 'ENABLE', any trailing empty partitions present at the successful completion of REORG are removed, including LOB

Function Level	New Feature	Comments
		tablespaces and auxiliary indexes.
V12R1M507	Deletion of old statistics when using statistics profiles	**Unavoidable change.** If you specify USE PROFILE with RUNSTATS or with inline statistics for REORG or LOAD, Db2 deletes existing statistics that are not part of the profile.
V12R1M509	Tamper-proof audit policies	To avoid this, do not specify 'T' for the DB2START column in catalog table SYSIBM.SYSAUDITPOLICIES

Incompatible/Unavoidable Db2 12 for z/OS Function Level Changes

This article discusses the new features and functions introduced by Db2 12 for z/OS function levels which are either incompatible changes or in an unofficial category called "unavoidable changes". Be aware that this is not official IBM documentation nor is it endorsed by IBM.

V12R1M503 Temporal Auditing

This is an incompatible change that affects the result set when retrieving rows from system-period temporal tables with ON DELETE ADD EXTRA ROW attribute that have been altered to add a DATA CHANGE OPERATION column. When you add a DATA CHANGE OPERATION column (or *auditing column*) to a system-period temporal table, Db2 assigns null values to that DATA CHANGE OPERATION column in the existing rows.

Prior to function level V12R1M503, temporal queries run against these tables do not return history rows with null in the DATA CHANGE OPERATION column, which means that the data returned does not include the expected historical data.

From function level V12R1M503 onwards, associated history rows with null in the DATA CHANGE OPERATION column *are* included in the result set for system-period temporal queries. Rows added to a history table as a result of the ON DELETE ADD EXTRA ROW attribute are not included.

Who Is Affected?

This only applies to queries against system-period temporal tables which have been altered to add a DATA CHANGE OPERATION column and where the ON DELETE ADD EXTRA ROW attribute was defined before function level V12R1M503.

To detect whether you are affected by this change, start a trace to collect IFCID 0376 records. If field QW0376FN in the trace output contains 1215031, then result sets for queries with period specifications will include rows from the history table with a null in its DATA CHANGE OPERATION column and you need to assess the impact.

Mitigating Actions

If you have coded your application to handle the missing history rows or you expect them to be missing, then ***you need to modify your application*** to handle the inclusion of the history rows in the result set for system-period temporal queries.

V12R1M503 System Time Override Stored Procedure No Longer Supported

This is an incompatible change that affects Db2 users who ***replicate system-period temporal tables***. The system time override stored procedure SET_MAINT_MODE_RECORD_NO_TEMPORALHISTORY is not intended for general use but for use by replication products. If your replication product runs

with APPLCOMPAT V12R1M502 or lower, then the stored procedure is supported, but not if the product runs with a higher APPLCOMPAT.

SET_MAINT_MODE_RECORD_NO_TEMPORALHISTORY is effectively replaced by a new global variable, SYSIBMADM.REPLICATION_OVERRIDE, which is part or the solution for system-period temporal tables that improves on the existing but limited replication capability by enabling replication of the audit columns.

Who Is Affected?

This only applies to users who replicate system-period temporal tables. To detect whether you are affected, start a trace to collect IFCID 376 records. In the trace output, function code 1215032 or 1203 in field QW0376FN indicates which products or programs are calling the stored procedure.

Mitigating Actions

If you are affected, contact your replication product vendor(s) to discuss available product updates and to obtain guidance on APPLCOMPAT settings.

V12R1M505 Reduced Latency Between Db2 for z/OS and IDAA

This is an unavoidable change that only applies to Hybrid Transactional Analytical Processing (HTAP) environments. It is not so much an unavoidable change as a ***performance improvement*** which is intended to reduce the latency between Db2 for z/OS and the IBM Db2 Analytics Accelerator (IDAA) where changes to Db2 data are propagated to IDAA with high frequency using IBM Integrated Synchronization. This is a non-functional change.

Who Is Affected?

Db2 for z/OS users with IDAA V7.5 or higher who exploit Integrated Synchronisation for HTAP capabilities.

Mitigating Actions

Test your HTAP applications in test/pre-production to make sure that there is no performance regression and that they are functionally identical to pre-function level V12R1M505.

V12R1M505 Package REBIND Phase-in

This is an unavoidable change. Rebind phase-in allows Db2 to rebind a package concurrently with the package being executed. From function level V12R1M505 onwards, rebind operations create a new – additional – copy of the package. New threads use the new copy, existing threads continue to use the old copy that is being phased-out. REBIND SWITCH PREVIOUS/ORIGINAL also benefits from rebind phase-in. This is an availability improvement that reduces the disruptive impact of the REBIND process. It is dependent on the COPYID columns that are added when the catalog is updated to catalog level V12R1M505.

Who Is Affected?

Everyone who uses REBIND.

Mitigating Actions

Test your rebind processes in test/pre-production. There is no effect on SQL DML, DDL or DCL behaviour. If your existing rebind procedures require you to take an application outage, you can forego this testing until you are ready to try to avoid the outage by exploiting rebind phase-in.

V12R1M505 RUNSTATS Sampling and Real-Time Statistics

This is an incompatible change that alters the behaviour of the default value for system parameter STATPGSAMP, which controls whether the RUNSTATS utility uses page-level sampling for universal table spaces by default. Prior to function level V12R1M505, the default STATPGSAMP value of SYSTEM has the same meaning as YES. From function level V12R1M505 onwards, SYSTEM has the same meaning as NO. In addition, if you use page-level sampling with RUNSTATS, either by default or explicitly, it no longer updates real-time statistics (RTS).

Who Is Affected?

Everyone who uses the RUNSTATS utility and has system parameter STATPGSAMP set to SYSTEM.

Mitigating Actions

There are two possible actions to avoid page sampling and ensure RTS is updated:

- Set system parameter STATPGSAMP=NO, and don't specify SAMPLE with RUNSTATS.
- If you accepted the STATPGSAMP default, specify TABLESAMPLE SYSTEM NONE without the SAMPLE option in the RUNSTATS statement.

V12R1M506 Implicit Dropping of Explicitly-created Tablespaces

This is an incompatible change that is part of the ease-of-use changes for the dropping of tables in universal tablespaces (UTS). The incompatibility is that dropping an auxiliary table in an explicitly created LOB table space implicitly drops the LOB table space.

Who Is Affected?

There are two cases where Db2 for z/OS users can be affected:

- Users expecting an auxiliary (LOB) tablespace to be available for reuse after dropping an auxiliary table.
- Users dropping an auxiliary (LOB) tablespace after dropping the associated auxiliary table.

Mitigating Actions

If you manage LOB tablespaces in-house with your own tools/procedures, you might need to modify those to accommodate the implicit dropping of the auxiliary tablespace when the auxiliary table is dropped.

The same applies to vendor-tooling that generates DDL statement work flows. You may need to reconfigure the tooling or contact the vendor to make sure it is compatible with function level V12R1M506 functionality.

V12R1M507 Deletion of Old Statistics When Using Statistics Profiles

This is an unavoidable change. When you activate function level V12R1M507 or higher and specify USE PROFILE with RUNSTATS or for in-line statistics collection with REORG or LOAD, Db2 deletes any old statistics that are not part of the profile. This is to ensure that the Optimizer doesn't use conflicting statistics when determining SQL statement access paths.

Who Is Affected?

Db2 for z/OS users of statistics profiles.

Mitigating Actions

In general, it is recommended that you ***do not take any mitigating actions***. If you use statistics profiles for tables where you want to keep old statistics in the catalog, you have two options:

- Save a copy of the old statistics before re-collecting statistics with USE PROFILE and insert those statistics back into the catalog after the utility has completed.
- Stop using statistics profiles for those tables and specify all the statistics options in the collecting utility job.

Function Level Activation Recommended Practices

To help Db2 for z/OS professionals use the information provided in the preceding sections effectively, this section provides some recommended practices, in three main topics:

- Recommended practices for function level activation testing.
- Recommended practices for advancing to Function Level FL500/FL501.
- Recommended practices for advancing beyond Function Level FL500/FL501.

The recommended practices for advancing Function Levels are based on the assumption that you plan to avoid exploiting new features and functions but can be adapted for the cases where you do plan exploitation. They are also designed with a defensive approach in mind, to minimise service disruption. They do not include any recommendations for the maintenance roll-out process. It is assumed that you will continue with your existing maintenance roll-out strategy.

Recommended Practices for Function Level Activation Testing

One of the inhibitors holding back function level activation is the perception that it is like a full release or mini-release migration and requires the same level of testing, including comprehensive application regression testing.

Because you can effectively switch off system features and control the introduction of SQL features through APPLCOMPAT, the testing required for function level activation can be regarded as more like that for a maintenance roll-out. The recommendation for function level activation is to perform the same level of testing as you currently do for a maintenance roll-out, with one caveat. If you are affected by any of the incompatible or unavoidable changes, then you should modify your testing schedule to test for processes and applications affected by those changes.

Recommended Practices for Advancing to Function Level FL500/FL501

Advancing to Function Level FL500 or FL501 is unlike any other Function Level activation process because it is a required step before you can run any further CATMAINT processes. FL100 supports fall-back to Db2 11 for z/OS NFM, which doesn't tolerate a catalog level beyond V12R1M500. To ensure the integrity of the fall-back process, FL100 prohibits CATMAINTs which would prevent fall-back to Db2 11.

The recommended practice is as follows:

- Confirm you have no contingency plans to fall-back to Db2 11 for z/OS NFM.
- Be at the highest possible maintenance level that is compatible with your maintenance strategy.
- Run with the new maintenance level in production for at least 1 – 2 months prior to activating FL500/FL501.
- If you are planning to exploit any of the new non-SQL features, expand your testing process to include these.
- Ensure all other non-SQL features are turned off.
- Choose a quiet time to activate the new function level. Remember, this does not require a Db2 restart.

    - Prior to activating the function level, use the -ACTIVATE FUNCTION LEVEL … TEST command to validate all the required conditions for function level activation have been met.
    - Activating a function level using the -ACTIVATE FUNCTION LEVEL command is not a disruptive process. Doing this during a quiet time is purely defensive in order to improve confidence in the process.

Recommended Practices for Advancing Beyond Function Level FL500/FL501

Once you have activated FL500 or FL501, you are now in a position to execute additional CATMAINT utilities and to advance function levels.

The recommended practice is as follows:

- Be at the highest possible maintenance level that is compatible with your maintenance strategy.
- Run with the new maintenance level in production for at least 1 – 2 months prior to running any required CATMAINTs or activating higher function levels. The point has already been made but is worth repeating, that your maintenance level must support the highest ever function level activated for the subsystem or data sharing group. Therefore, do not activate a function level that is the same level as the new maintenance level until you are sure you will not back out the new maintenance level.
    - To combine upgrading the maintenance level and the function level, consider activating the highest function level that is supported by your old maintenance level at the same time as the new maintenance level roll out.
- To minimise disruptive changes, ***run CATMAINT to take you to the highest possible catalog level given your maintenance level***.
    - The last Db2 12 Catalog Level, V12R1M509, requires maintenance level V12R1M509, available from February 2021.
    - The previous Db2 12 Catalog level, V12R1M507, requires maintenance level V12R1M507, available from June 2020.
- Run with the new Catalog Level for at least 1 month prior to activating any given function level.
- If you are planning to exploit any of the new non-SQL features, expand your testing process to include these.
- Ensure all other non-SQL features are turned off.
- Choose a quiet time to activate the new function level. Remember, this does not require a Db2 restart.
    - Prior to activating the function level, use the -ACTIVATE FUNCTION LEVEL … TEST command to validate all the required conditions for function level activation have been met.
- Activating a function level using the -ACTIVATE FUNCTION LEVEL command is not a disruptive process. Doing this during a quiet time is purely defensive in order to improve confidence in the process.

Managing Application Compatibility

For applications to exploit new features and functions introduced by advancing Function Levels, they must be bound using an APPLCOMPAT setting that supports that functionality. This section provides guidance on three broad areas:

- Static SQL packages.
- Dynamic SQL packages used locally.
- Dynamic SQL packages used by remote applications accessing Db2 via TCP/IP and DRDA.

APPLCOMPAT and Static SQL Packages

There is no recommendation to rebind existing static SQL packages to use a higher APPLCOMPAT setting, ***as there is no benefit*** whatsoever. Doing so could even produce unexpected results if there are any unresolved SQL incompatibilities resulting from a change in behaviour.

However, if you want to modify applications to use new SQL capabilities, you must resolve any and all incompatibilities in each affected application program, and when you bind the application (with BIND ADD or BIND REPLACE) you must specify the APPLCOMPAT level that supports the new features. It is recommended that you do so 1 – 2 months after function level activation, to make sure the new function level is stable.

SQLLEVEL

The precompiler option SQLLEVEL specifies the Function Level (or release) to be used to specify the SQL syntax allowed by the Db2 pre-compiler or coprocessor during the program preparation process. It has an equivalent system parameter, SQLLEVEL, which specifies the default value used if none is specified at program preparation time.

It is recommended that you set the SQLLEVEL system parameter to your current function level once you are satisfied the new function level is stable. This will allow program preparation using new features and functions. Nevertheless, for an application program to be able to execute using the new functionality, it still has to be bound with the appropriate APPLCOMPAT attribute.

Your procedures might require the SQLLEVEL system parameter to be set at a lower function level, but you will need controls in place, firstly to prevent application programmers from overriding the system parameter, and secondly to allow exceptions where applications are required to exploit new features.

APPLCOMPAT and Locally Executed Dynamic SQL Packages

Locally executed dynamic SQL packages are used for a wide range of purposes, including:

- Vendor tools such as catalog navigation and schema change management tools.
- Executing DDL to implement schema change.
- Executing ad-hoc SQL queries.
- Executing SQL DML for purposes such as data correction or application prototyping.

You will therefore need to have a flexible approach to setting APPLCOMPAT, according to your requirements in each case. For example, many vendor tools will use the CURRENT APPLICATION COMPATIBILITY special register to control application compatibility, so you will need to bind these vendor packages using the APPLCOMPAT attribute recommended by the vendor.

On other cases, you may decide you need multiple copies of the packages, in separate collections, with different APPLCOMPAT settings. This is particularly true of the DDL packages. DDL packages bound with an APPLCOMPAT of V12R1M504 or higher will be unable to create segmented or classic partitioned tablespaces, so if you have a need to create or recreate any of these, you will need to keep a copy of your DDL package at APPLCOMPAT V12R1M503 or lower.

You can execute those different package versions by creating a plan for each required package / APPLCOMPAT attribute combination, and referencing that plan name on, for example, the RUN subcommand under the DSN subcommand processor.

If you plan to use the CURRENT APPLICATION COMPATIBILITY special register, remember that the package must be bound with an APPLCOMPAT attribute at least as high as the highest special register setting you plan to use. This is a change from Db2 11, where you could use the CURRENT APPLICATION COMPATIBILITY special register to specify a value as high as the current Function Level, even if it's higher than the package APPLCOMPAT.

APPLCOMPAT and Dynamic SQL Packages for DRDA Client Applications

Most DB2 sites will want to selectively move DRDA client applications to a higher APPLCOMPAT setting, to be able to limit the amount of testing to do, and to be able to introduce the new functionality in a carefully controlled way. This paper refers to the packages in the NULLID collection, but if you have other such dynamic SQL packages the same principles apply.

This paper discusses three ways of managing the APPLCOMPAT bind attribute for the NULLID packages. At least two of these require additional copies of the NULLID packages with different APPLCOMPAT attributes in new collections. It is recommended that you use meaningful names for the collections such as NULLID_V12R1M502 and NULLID_V12R1M505, to match the package APPLCOMPAT attributes.

Once the new collections and package copies have been created, you will need to be able to direct specific applications or application programs to use the required collection/package combination This paper discusses three ways of doing:

- Setting the CURRENT APPLICATION COMPATIBILITY special register
- Configuring the client to direct applications to the desired collection by defining multiple data sources at the client – client-side configuration.
- Configuring DB2 for z/OS to selectively direct applications to different collections using system monitoring profiles – server-side configuration.

Setting the CURRENT APPLICATION COMPATIBILITY special register

To avoid using multiple collections, rebind all the NULLID packages, setting APPLCOMPAT to the highest level you plan to use for all applications. This is because the setting of the CURRENT APPLICATION COMPATIBILITY special register is limited by the package APPLCOMPAT attribute, not the current Function Level. This approach is not recommended for two reasons:

- It would require that you retest all your distributed applications to make sure there was no unexpected change in behaviour and that all inconsistencies had been resolved.
- It requires you to set the CURRENT APPLICATION COMPATIBILITY special register in the application for those applications that require a lower application compatibility attribute than the package.

Client-side Configuration

Client-side configuration involves creating multiple data sources at each client, one for each different collection that will be used from that client, and then getting applications to open a logical connection to the correct data source. A given data source can be associated with a specific collection by using the currentPackageSet or currentPackagePath client configuration option.

There are many ways of configuring clients to use data sources. For example, see the IBM documentation at "Getting started with IBM Data Server Drivers", "Common IBM Data Server Driver for JDBC and SQLJ properties for Db2 servers" and "Connecting to databases for ODBC and CLI".

However, this is not a straightforward solution especially if you've got dozens, or hundreds, or even thousands of clients to configure, with many distinct applications connecting to a single Db2 subsystem or data sharing group.

Server-side Configuration

The main alternative to client-side configuration is server-side or DB2-side configuration. Although not without its challenges, the advantage of server-side configuration is that much of the necessary configuration is done in one place, using system profiles.

System profiles were introduced as long ago as DB2 9, and have been successively enhanced in every release since then. They provide a range of capabilities, including setting ZPARMs at a more granular level, production modelling in test systems, and managing application-specific behaviour via APPLCOMPAT, all using online change. Configuration information for system profiles is stored in two tables, SYSIBM.DSN_PROFILE_TABLE and SYSIBM.DSN_PROFILE_ATTRIBUTES, and managed using the -START PROFILE and -STOP PROFILE commands.

SYSIBM.DSN_PROFILE_TABLE is used to define the filtering criteria – what is being monitored– and SYSIBM.DSN_PROFILE_ATTRIBUTES is used to define the attributes that are applied to what is being monitored.

Managing NULLID package APPLCOMPAT levels using system profiles is fairly complicated so therefore this paper goes into some detail about how to do this. It involves setting special register values for remote applications, and in particular, setting the CURRENT PACKAGE PATH special register.

First you need to create filters that identify which applications you want to direct to those collections with different APPLCOMPAT attributes – that is, you need to create one or more profiles for those applications. Each row in SYSIBM.DSN_PROFILE_TABLE defines a single profile, with a unique identifier in column PROFILEID. The eligible filters for setting special registers are listed below divided into categories, with only one category of filter allowed per profile:

1. LOCATION
2. PRDID
3. AUTHID, ROLE
4. COLLID, PKGNAME
5. CLIENT_APPLNAME
6. CLIENT_USERID
7. CLIENT_WRKSTNNAME

If you create multiple profiles with different filtering categories that a thread or connection could match on, then you need to take into account the DB2 rules for applying multiple matching profiles, and the order of precedence.

Once you've created the required profiles, you need to insert a row with a matching PROFILEID value into SYSIBM.DSN_PROFILE_ATTRIBUTES to set the CURRENT PACKAGE PATH special register for each profile. For example, to ensure an application matching the profile filtering runs with APPLCOMPAT V12R1M505, insert a row with the correct PROFILEID value into SYSIBM.DSN_PROFILE_ATTRIBUTES,

with the value “SPECIAL REGISTER” in the KEYWORDS column, and the value “SET CURRENT PACKAGE PATH = ‘NULLID_V12R1M505’” in the ATTRIBUTE1 column.

To make the profile effective, you need to set the PROFILE_ENABLED column in DSN_PROFILE_TABLE for that profile to ‘Y, and you need to activate it by issuing the -START PROFILE command.

If you need granular APPLCOMPAT attribute control at the application or application program level, then you can consider filtering on the CLIENT_APPLNAME, CLIENT_USERID or CLIENT_WRKSTNNAME which are provided in the client information fields. However, these must either be coded in the application or set via client-side configuration.

When you issue -START PROFILE, all profiles where PROFILE_ENABLED is set to ‘Y’ are activated. You can’t start and stop individual profiles via the PROFILEID identifier. If you are data sharing, you need to coordinate these commands across the data sharing group, as the -START and -STOP PROFILE commands have member scope.

Managing the APPLCOMPAT level of applications that use the NULLID packages is not straightforward. You will need to choose the strategy – client-side or server-side configuration – that best suits your requirements. If you cannot manage the APPLCOMPAT level of the applications that use the NULLID packages using client-side or server-side configuration, then you will need to resolve all inconsistencies and rebind the NULLID packages with a higher APPLCOMPAT setting deploying applications that require the higher APPLCOMPAT setting.

Appendix A – The Continuous Delivery Mechanism

With Continuous Delivery, there is one single delivery stream for both defect fixes and enhancements using standard PTFs (and collections of PTFs like PUT levels and RSU packages).

The DB2 continuous delivery mechanism consists of four levels:

- The Maintenance Level (ML), which is raised by applying maintenance. This is sometimes also known as code level. The maintenance provided by IBM contains defect fixes and new enhancement fixes in the same maintenance package. Most new functions are shipped disabled until the appropriate new Function Level is activated
- The Catalog Level (CL), which is the vehicle to enable new Function Levels. Advancing the Catalog Level is via CATMAINT. This is cumulative and skip level is possible, as CATMAINT checks the current Catalog Level and makes sure all necessary CATMAINT changes are applied in the correct order. Catalog changes are needed for some Function Levels but not all. This is clearly documented in the activation instructions for each Function Level.
- The Function Level (FL) introduces new DB2 for z/OS features and functionality. Function levels need to be activated and are not available until activation. They are also cumulative and skip level is possible. When a Function Level is activated, there is no impact to the application or any change in existing application behaviour. Many of these features are related to SQL DML, DDL, DCL and XML syntax which are controlled by the APPLCOMPAT level, but some features and functions are not related to the APPLCOMPAT level and are introduced when you activate the associated Function Level.
    - It's important to note here that while the Maintenance Level can be advanced on a member-by-member basis, advancing the Catalog Level and Function Level both apply to the entire data sharing group. The sections below describe the relationships between Maintenance Level, Catalog Level, and Function Level.
- APPLCOMPAT level (AC), which is set at the application (package) level. This provides an "island of stability" for applications which have an APPLCOMPAT setting lower than the current DB2 Function Level. APPLCOMPAT determines the Function Level that applications can exploit and can be set on a package by package basis. An application package can be rebound to advance to a higher APPLCOMPAT level or can be rebound to fall back to a lower one. The golden rule is that, for an application to take advantage of new features introduced in, for example, Function Level V12R1M505, the application must be bound with APPLCOMPAT(V12R1M505) or higher, and the Function Level must be V12R1M505 or higher at the time the package is bound. The APPLCOMPAT level specified when a package is bound or rebound must be less than or equal to the current Function Level and takes precedence over the Function Level. Once an application is bound with a given APPLCOMPAT, this ensures that any statements exploiting new SQL syntax continue

to work even if the Function Level is later moved back to an earlier level. The default setting for APPLCOMPAT is specified as a system parameter (ZPARM), which allows you to hold back the default application compatibility level or advance it as your strategy requires. When APPLCOMPAT is explicitly specified on a BIND or REBIND command, this overrides the system default. This in turns allows some applications to run at a lower or higher APPLCOMPAT setting than the system default.

There are some additional key points to remember:

- The CATMAINT job run to migrate from DB2 11 NFM to DB2 12 sets the Catalog Level at V12R1M500, even though the Function Level is V12R1M100.
- DB2 11 tolerates Catalog Level V12R1M500 in the event of fall back from DB2 12 to DB2 11 NFM.
- APPLCOMPAT V12R1M100 is functionally equivalent to V11R1.
- The minimum starting point for continuous delivery is DB2 12 Function Level V12R1M500.

Understanding New Function Levels

New Function Levels are introduced in the maintenance stream, alongside the corrective maintenance. Some require the catalog and directory to be updated via CATMAINT, which raises the Catalog Level, and some do not. The activation of a Function Level implies the activation of all lower Function Levels.

IBM provides a DB2 Function Levels landing page: https://www.ibm.com/support/knowledgecenter/SSEPEK_12.0.0/wnew/src/tpc/db2z_db2functionlevels.html . This provides a brief description of the new features/functions included in the Function Level and also identifies the required APAR(s). There is a link to a dedicated page for each Function Level which provides detailed information about each new feature/function introduced in the Function Level, plus the following key items of information; any migration process changes to support the new Function Level such as jobs, panels and CLISTs; activation details (for example, if CATMAINT is required); and the required Catalog Level.

The Relationship Between Maintenance Level and Function Level

The Function Level that a DB2 subsystem or data sharing group can support is dependent on the Maintenance Level of that subsystem or group. All members of a data sharing group must be at the Maintenance Level required to support a Function Level before that Function Level can be activated. Bear in mind that Maintenance Level has member scope, whereas Function Level activation has group scope.

Activating a Function Level prevents backing off the Maintenance Level required by that Function Level. It is possible to return to a lower Function Level – for example, if you are at Function Level V12R1M506, you can return to V12R1M504* (the '*' indicates that the subsystem or group has previously been at a higher Function Level). If you return to a lower Function Level, all existing usage of a feature introduced by a Function Level will continue to be allowed, but new exploitation of the returned-from Function Level will be prevented.

Returning to a previous Function Level requires that the Maintenance Level of all group members or of the standalone subsystem must still support the highest Function Level ever activated.

The Relationship Between Catalog Level and Maintenance Level/Function Level

Just like the Function Level, the Catalog Level is dependent on the Maintenance Level. All members of a data sharing group must be started with the Maintenance Level that supports the Catalog Level. For example, when CATMAINT has been run to bring the Catalog Level up to V12R1M502, all members must be started with DB2 12 Maintenance Level 502. For a standalone subsystem, that subsystem must be started with the Maintenance Level that supports the Catalog Level. Running CATMAINT prevents backing off any maintenance required by that Catalog Level.

The Function Level is dependent on the Catalog Level as well as on the Maintenance Level – for example:

- Function Level V12R1M502 activation includes a required CATMAINT to take the Catalog Level to V12R1M502.
- Function Level V12R1M504 activation does not include a CATMAINT but requires the Catalog Level to be V12R1M503, the highest Catalog Level prior to Function Level V12R1M504.

This has not yet been the case, but a CATMAINT might require that a particular Function Level is activated before the catalog change occurs, for preconditioning.

The Change in Strategy for APPLCOMPAT Introduced by DB2 12

In DB2 12, APPLCOMPAT is extended to support Function Levels (for example, V12R1M501) as well as release levels (for example, V11R1). APPLCOMPAT now has to support many Function Levels. Additionally, APPLCOMPAT is extended to support DDL and DCL as well as DML and XML. This means that in DB2 12 DDL and DCL are sensitive to APPLCOMPAT, not the Function Level.

You must therefore still ***rebind a package*** with the required APPLCOMPAT level in order ***to exploit new*** DML, DDL, DCL, and XML function. Packages must be bound with an APPLCOMPAT less than or equal to the current Function Level.

There is ***no need to rebind all packages with a new, higher APPLCOMPAT level***. However, it is still the recommended best practice to regularly rebind all packages, even without changing the APPLCOMPAT level, to:

- Benefit from the latest run time performance improvements.
- Gain exposure to new access path selection improvements.
- Benefit from fixes for defects.
- Reduce exposure to latent issues created previously under earlier releases of DB2 or earlier Maintenance Levels.

With regard to the setting of the APPLCOMPAT property for a package, BIND REPLACE does not reuse any bind option from the existing package if the option is not explicitly specified. The SQL statements can be totally different so BIND REPLACE is therefore considered a new bind. BIND REPLACE and BIND ADD will use the APPLCOMPAT ZPARM value if the APPLCOMPAT BIND option is not specified.

On the other hand, REBIND and BIND COPY are the only subcommands that ***reuse the existing options if APPLCOMPAT is not explicitly specified***. However, APPLCOMPAT is only a 'sticky' option if the APPLCOMPAT column in the SYSIBM.SYSPACKAGE catalog is populated for the package. If not, the APPLCOMPAT ZPARM value will be used. Therefore, you should be careful in advancing the level-id of the APPLCOMPAT ZPARM.

With regard to the CURRENT APPLICATION COMPATIBILITY Special Register, the value has to be less than or equal to the APPLCOMPAT level of the executing package, independent of the current DB2 Function Level. This is a change from DB2 11 where the value can be greater than the APPLCOMPAT level of the executing package (but not higher than the DB2 release level).

Appendix B – Survey Results

This appendix provides comprehensive survey results, including those not already discussed in the key survey results. Some of the charts are repeated from the main document but are included here for convenience and completeness.

The data on which the Catalog Level chart Is based either contains form-filling errors or indicates a misunderstanding of catalog levels in the user community. This is because there is data for catalog levels V12R1M501, V12R1M506 and V12R1M508, which don't exist, but none for V12R1M509 which is required for FL509.

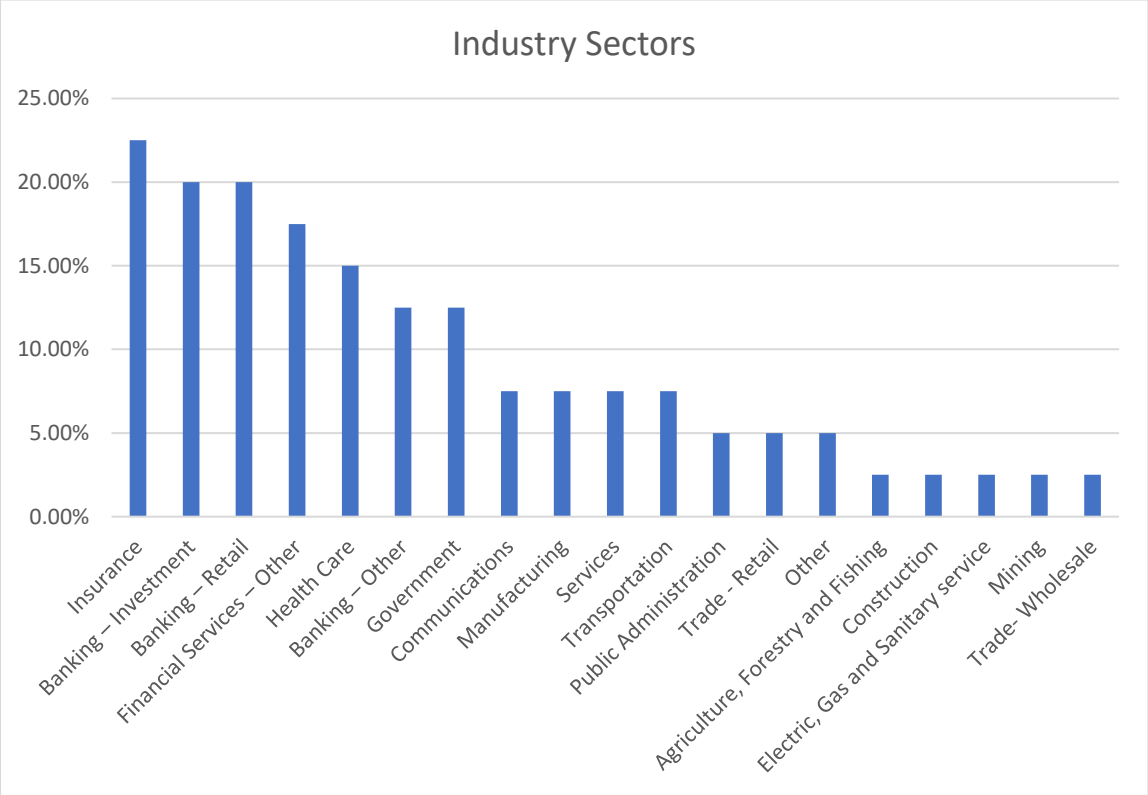

Figure 4 Industry Sectors

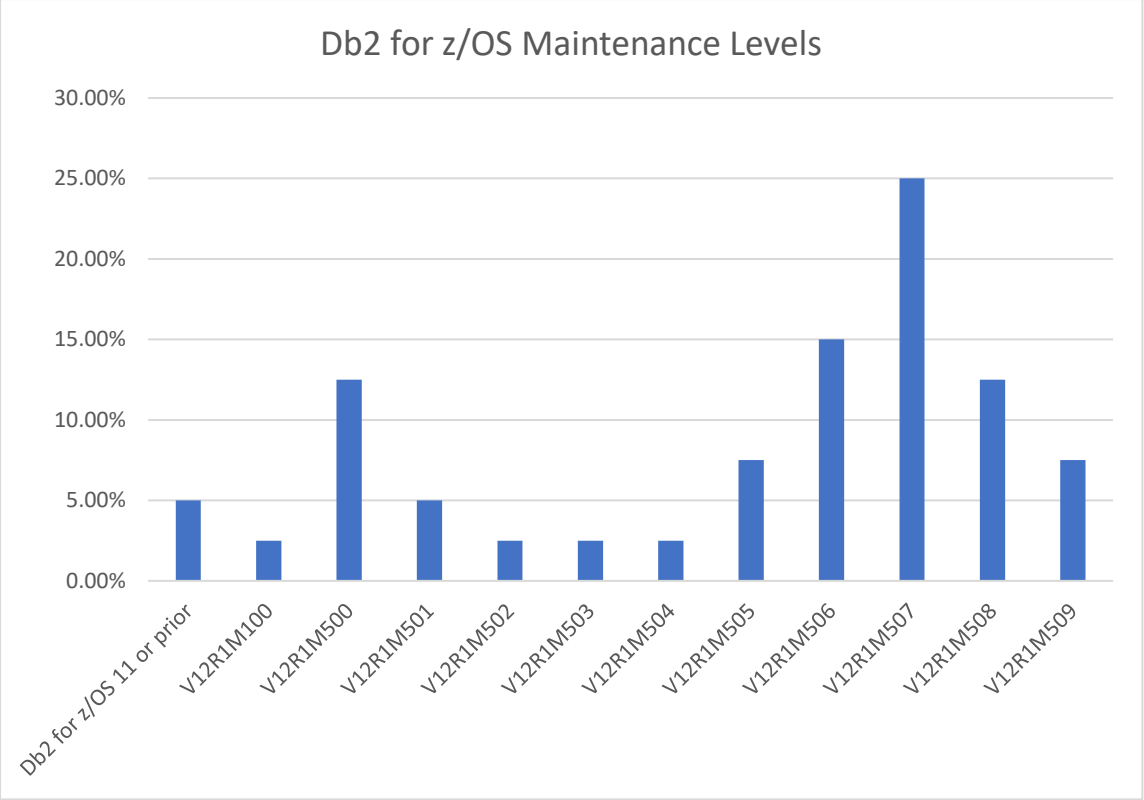

Figure 5 Db2 for z/OS Maintenance Levels

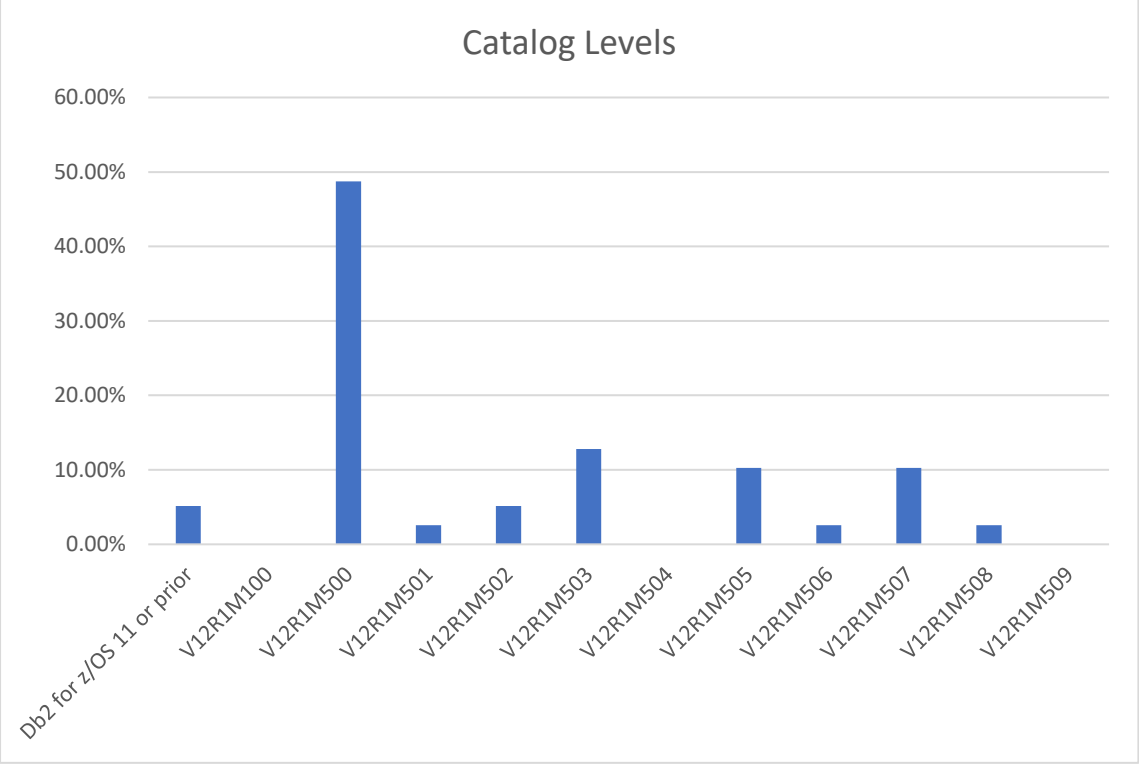

Figure 6 Db2 for z/OS Catalog Levels

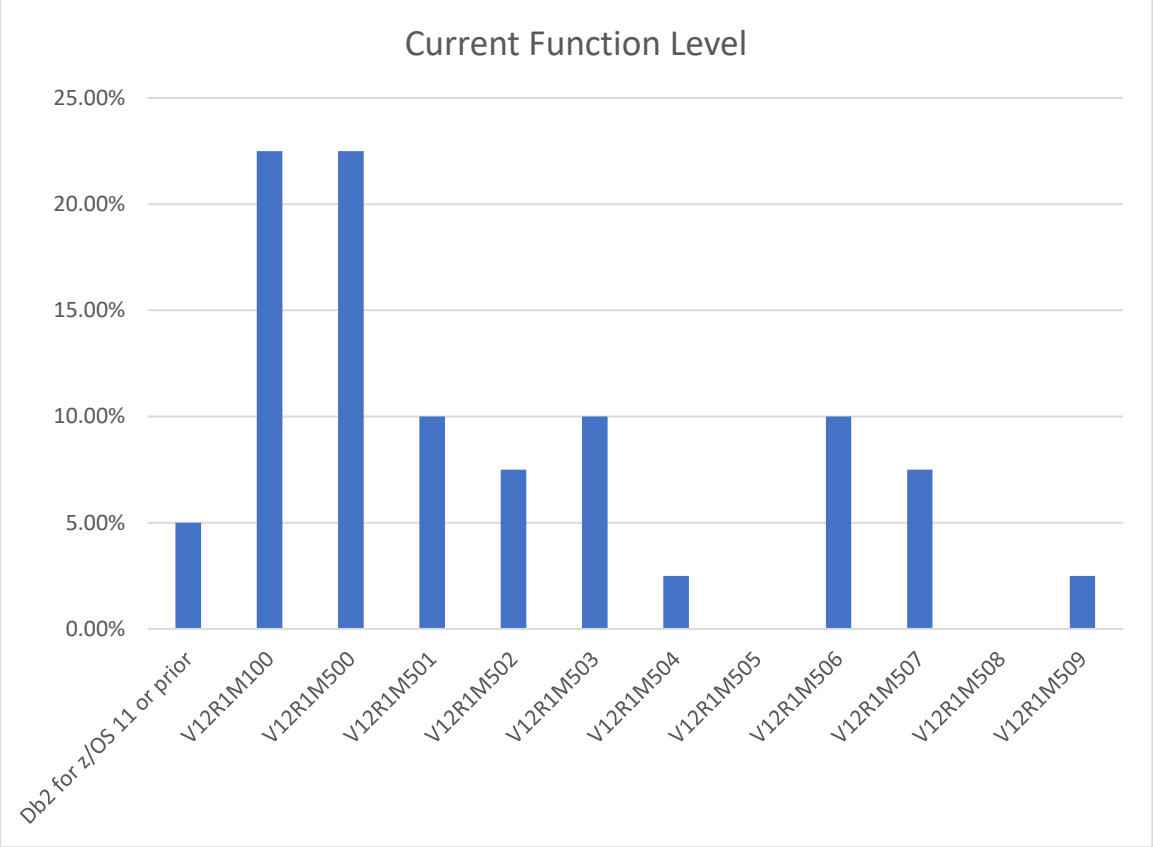

Figure 7 Current Function Level

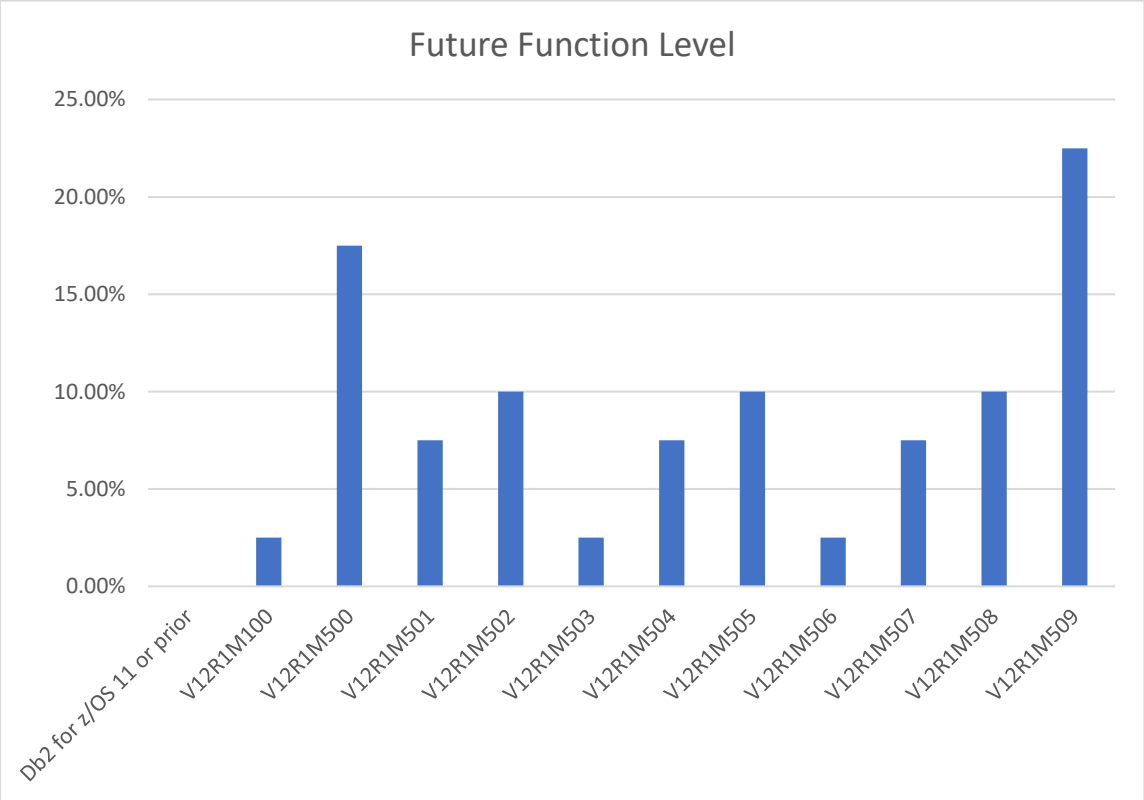

Figure 8 Planned Function Level

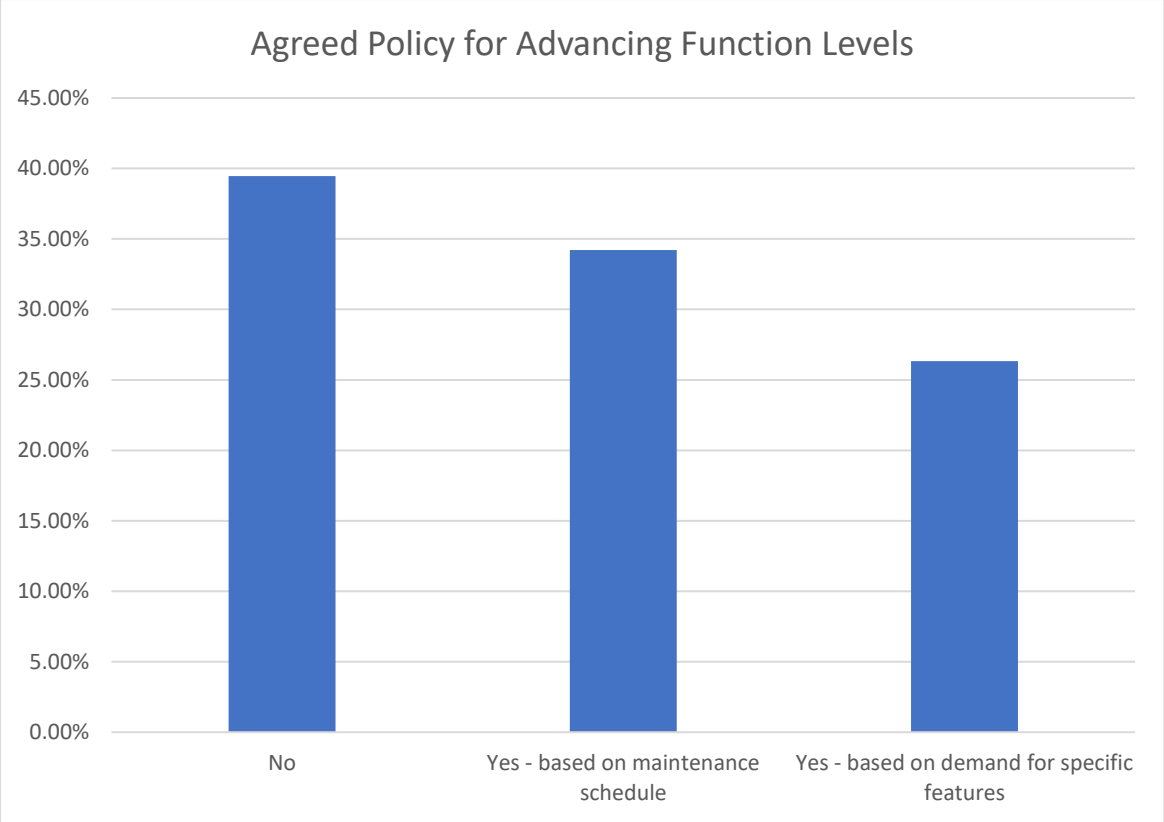

Figure 9 Agreed Policy for Advancing Function Levels

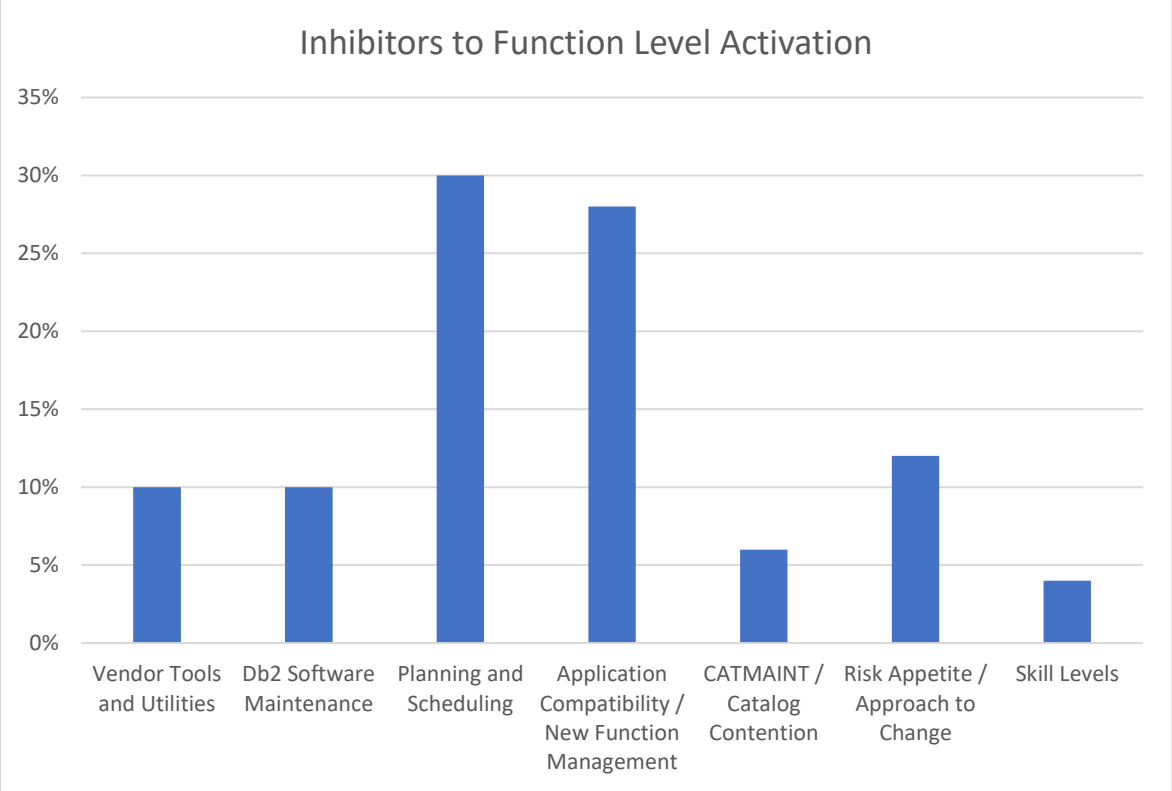

Figure 10 Inhibitors to Function Level Activation