DB2 12 — The ultimate enterprise database for business-critical transactions and analytics

DB2 for z/OS: Continuous Delivery of New Features (part 2)

Chris Crone – DE DB2 Development
Presented by Mark Rader
WSC: DB2 for z/OS
Applications
Static SQL, DDL, and DCL

- In DB2 11, Static SQL is controlled by APPLCOMPAT BIND option

- In DB2 11, DDL and DCL is controlled by CM/NFM

- DB2 12 needs to take into account function levels changing more often and the need to control applications’ use of new function across one or more function levels
  - In DB2 12, the APPLCOMPAT BIND option is extended to support function levels (e.g. V12R1M501)
  - In DB2 12, the APPLCOMPAT BIND option is extended to support DDL and DCL, in addition to DML
Dynamic SQL, DDL, and DCL

- In DB2 11, by default, Dynamic SQL executed under the APPLCOMPAT Bind Option
  - Could be V10R1 or V11R1
    - SET CURRENT APPLICATION COMPATIBILITY could be used to SET to V11R1 if in NFM
- DB2 12 needs to take into account function levels changing more often and the need to control applications use of new function across one or more function levels
  - In DB2 12 the APPLCOMPAT BIND option is extended to support function levels (e.g. V12R1M501) for both STATIC and DYNAMIC SQL, DDL and DCL
  - The APPLCOMPAT BIND OPTION controls the MAX value that can be specified in SET CURRENT APPLICATION COMPATIBILITY
  - An APPLCOMPAT level will remain active even if the FL is lowered to a * mode (e.g. M501->M500*)
    - REBIND of the current APPLCOMPAT value is allowed even if the value exceeds the current *mode FL
    - SET CURRENT APPLICATION COMPATIBILITY continues to be allowed up to the value specified on BIND/REBIND for APPLCOMPAT
New Global Variables

- **PRODUCTID_EXT**
  - Contains the extended product identifier of the database manager that invoked the function.
  - The format of the extended product identifier values is `pppvvrrmmm`, where `ppp` is a 3-letter product code (such as `DSN` for DB2®), `vv` is the version, `rr` is the release, and `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 higher).

- **CATALOG_LEVEL**
  - Contains the level of the current catalog level.
  - The format of the catalog level values is `VvvRrMmmm`, where `vv` is the version, `r` is the release, and `mmm` is the modification level (such as '100', '500', '501'). For example, `V12R1M500` identifies DB2 12 after the activation of DB2 12 new function (function level 500 or higher).

- **DEFAULT_SQLLEVEL**
  - Contains the value of the default value of the SQLLEVEL SQL processing option (DECPSQLLV).
  - The format of the default SQL level values is `V10R1`, `V11R1`, or `VvvRrMmmm`, where `vv` is the version, `r` is the release, and `mmm` is the modification level (such as '100', '500', '501'). For example, `V12R1M500` identifies DB2 12 after the activation of DB2 12 new function.

- **DB2 V11 can query the new built-in global variables. Their values will be:**
  - SYSIBM.PRODUCTID_EXT: DSN111500
  - SYSIBM.CATALOGLEVEL: V12R1M500
  - SYSIBM.DEFAULT_SQLLEVEL: NULL
IBM Data Server Drivers
- ODBC/CLI
- JDBC
New client property to control new function - clientApplCompat

- Any level of DB2 Connect drivers should work with DB2 V12, both before and after new function is activated with no behavior change.
- Data server clients and drivers must be at the following levels to exploit DB2 for z/OS function-level application compatibility of V12R1M501 or greater:
  - 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
- clientApplCompat lets you control the capability of the client when updated drivers ship changes to enable new server capability. You might want specific control of driver capability when:
  - 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
- clientApplCompat V12R1M500 is required to access DB2 12 Server capability shipped after GA at function levels beyond DB2 V12R1M500.
### clientApplCompat relationship to DB2 APPLCOMPAT

<table>
<thead>
<tr>
<th>Driver Level</th>
<th>clientApplCompat setting</th>
<th>APPLCOMPAT setting***</th>
<th>Application Capability</th>
</tr>
</thead>
<tbody>
<tr>
<td>V10.5 FP2(-)</td>
<td>n/a</td>
<td>V10R1</td>
<td>Applications may exploit DB2 10 functionality</td>
</tr>
<tr>
<td>V11</td>
<td>n/a</td>
<td>V11R1</td>
<td>Applications may exploit DB2 11 functionality</td>
</tr>
<tr>
<td>V11 FP1(+)</td>
<td>not provided</td>
<td>V12R1M100 V12R1M500</td>
<td>Applications may exploit DB2 11 functionality</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Applications may exploit DB2 V12R1M500 functionality</td>
</tr>
<tr>
<td>V10.5 FP2(-)</td>
<td>n/a</td>
<td>V12R1M501 V12R1M5xx</td>
<td>SQLCODE -30025 Function Level V12R1M501 has a dependency on clientApplCompat being set</td>
</tr>
<tr>
<td>V11</td>
<td>n/a</td>
<td></td>
<td></td>
</tr>
<tr>
<td>V11 FP1(+)</td>
<td>V12R1M500</td>
<td>V10R1</td>
<td>Applications may exploit DB2 10 functionality</td>
</tr>
<tr>
<td></td>
<td></td>
<td>V11R1</td>
<td>Applications may exploit DB2 11 functionality</td>
</tr>
<tr>
<td></td>
<td></td>
<td>V11R1M100 V12R1M501</td>
<td>Applications may exploit DB2 11 functionality</td>
</tr>
<tr>
<td></td>
<td></td>
<td>V12R1M5xx V12R1M5(zz-1)</td>
<td>Applications may exploit DB2 V12R1M5(zz-1) functionality</td>
</tr>
<tr>
<td>V11 FP1(+)</td>
<td>V12R1M501(+)</td>
<td>V12R1M500</td>
<td>SQLCODE -30025 clientApplCompat must not exceed the server APPLCOMPAT setting</td>
</tr>
<tr>
<td>Vyy Fpy ###</td>
<td>V12R1M5zz</td>
<td>V12R1M5zz</td>
<td>Applications may exploit DB2 V12R1M5zz functionality Future function that requires new DRDA data flow</td>
</tr>
</tbody>
</table>

***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
Data Server Drivers - Pre DB2 11 FP1*

- ACTIVATE NEW FUNCTION (V12R1M500)

DB2 V12R1M100

DB2 V12R1M500

DB2 V12R1M501

DB2 V12R1M501

DB2 V12R1M500 Catalog

DB2 V12R1M501 Libraries

FL M501 Libraries must remain active

Bind with APPLCOMPAT V10R1, V11R1, V12R1M100
SET CURRENT APPLICATION COMPATIBILITY V10R1, V11R1, V12R1M100

Bind with APPLCOMPAT V10R1, V11R1, V12R1M100, V12R1M500
SET CURRENT APPLICATION COMPATIBILITY V10R1, V11R1, V12R1M100, V12R1M500

No change in client behavior
Server Capability is limited to APPLCOMPAT(V12R1M500)

*Including any any Data Server Driver post DB2 11 FP1 if clientApplCompat = V12R1M500 is not specified

© 2017 IBM Corporation
Data Server Drivers - DB2 11 FP1(+)

No change in client behavior – No need to change clientApplcompat until a new DRDA flow is required.
clientApplCompat – control of future function

- DB2 12 only allows the new client behavior to control behavior when the APPLCOMPAT package at the server supports that level of client server function.
- We anticipate that at some point in the future, a new server feature will require a new clientApplCompat level.
- For example let’s say a new datatype is added in DB2 12 at DB2 function level DB2 V12R1M5zz. At that point, client-server applications wanting to exploit that new capability will need to have:
  - Packages bound at the server with APPLCOMPAT(V12R1M5zz)
  - clientApplCompat set to V12R1M5zz
- A clientApplCompat that exceeds the Server APPLCOMPAT bind option for the data server driver packages will result in:
  - -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 clientApplCompat as you increase the APPLCOMPAT of the data server driver packages – it will only be necessary when new client capability needs to be exploited.
- Best Practice – only change your clientApplCompat when necessary
SQLCA and GET DIAGNOSTIC impacts

- SQL CONNECT is enhanced to return the DB2 product ID and its current and functional level of the connected server in the generated SQLCA

- Information about the server is placed in the SQLERRP and SQLERRMC fields
  - Information in the SQLERRP field contains the server product information
    - It contains the following information:
      - ppp is the product identifier:
        - 'AQT' IBM® DB2 Analytics Accelerator for z/OS
        - 'ARI' DB2 Server for VSE & 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 The version identifier such as '11' for Version 11.
      - rr The release identifier such as '01'.
      - m The modification level.
  - Information in the SQLERRMC field contains the functional levels
    - First token contains the functional level of the connected DB2 (e.g. 'V12R1M506').

- Clients adding new APIs to retrieve the current DB2 functional level
DDF Commands and Messages

- DISPLAY LOCATION
  - Changed to show the functional level of each connected system

- DDF Location Statistics
  - Changed to show the functional level of each connected system

- DDF Accounting Header
  - Changed to show the functional level of the connected systems
What the Clients see after Connect

- CLI introduce a new infotype SQL_DBMS_FUNCTIONLVL for SQLGetInfo api that will return the function level/buildlevel.
- JCC introduce a new getDatabaseFunctionalLevel() API driver that will return a string.

- DB2 LUW returns build level (e.g. "n1604281900").
- DB2 Z returns function level (e.g. "V12R1M504").
Catalog changes
New catalog table SYSIBM.SYSLEVELUPDATES

A row will be inserted into this table during ACTIVATE FUNCTION LEVEL and CATMAINT UPDATE

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FUNCTION_LVL</td>
<td>VARCHAR(10)</td>
<td>Function level when this record is inserted</td>
</tr>
<tr>
<td>PREV_FUNCTION_LVL</td>
<td>VARCHAR(10)</td>
<td>Previous function level</td>
</tr>
<tr>
<td>HIGH_FUNCTION_LVL</td>
<td>VARCHAR(10)</td>
<td>Highest activated function level</td>
</tr>
<tr>
<td>CATALOG_LVL</td>
<td>VARCHAR(10)</td>
<td>Catalog level when this record is inserted</td>
</tr>
<tr>
<td>OPERATION_TYPE</td>
<td>CHAR(1)</td>
<td>Type of operation: ‘C’ for catalog change, ‘F’ for function level, ‘M’ for maintenance update (some form of special catmaint for instance)</td>
</tr>
<tr>
<td>EFFECTIVE_TIME</td>
<td>TIMESTAMP(12)</td>
<td>Time when a change in function or catalog level has completed.</td>
</tr>
<tr>
<td>EFFECTIVE_LRSN</td>
<td>CHAR(10)</td>
<td>RBA (or LRSN for data sharing) at the time when a change in function or catalog level has completed.</td>
</tr>
<tr>
<td>OPERATION_TEXT</td>
<td>VARCHAR(256)</td>
<td>The text of the operation.</td>
</tr>
<tr>
<td>GROUP_MEMBER</td>
<td>VARCHAR(24)</td>
<td>Name of the group member on which the operation was executed.</td>
</tr>
</tbody>
</table>
New catalog table SYSIBM.SYSLEVELUPDATES

<table>
<thead>
<tr>
<th>FUNCTION_LVL</th>
<th>PREV_FUNCTION_LEVEL</th>
<th>HIGH_FUNCTION_LEVEL</th>
<th>CATALOG_LVL</th>
<th>OPERATION_TYPE</th>
<th>EFFECTIVE_TIME</th>
<th>OPERATION_TEXT</th>
</tr>
</thead>
<tbody>
<tr>
<td>V12R1M100</td>
<td>V11R1M500</td>
<td>V12R1M100</td>
<td>V12R1M500</td>
<td>C</td>
<td>TIME 1</td>
<td>CATMAINT UPDATE UNLDDN V12R1M500</td>
</tr>
<tr>
<td>V12R1M500</td>
<td>V12R1M100</td>
<td>V12R1M500</td>
<td>V12R1M500</td>
<td>F</td>
<td>TIME 2</td>
<td>ACTIVATE FUNCTION LEVEL(V12R1M500)</td>
</tr>
<tr>
<td>V12R1M503</td>
<td>V12R1M500</td>
<td>V12R1M503</td>
<td>V12R1M500</td>
<td>F</td>
<td>TIME 3</td>
<td>ACTIVATE FUNCTION LEVEL(V12R1M503)</td>
</tr>
<tr>
<td>V12R1M500</td>
<td>V12R1M503</td>
<td>V12R1M503</td>
<td>V12R1M500</td>
<td>F</td>
<td>TIME 4</td>
<td>ACTIVATE FUNCTION LEVEL(V12R1M500)</td>
</tr>
<tr>
<td>V12R1M500</td>
<td>V12R1M503</td>
<td>V12R1M503</td>
<td>V12R1M505</td>
<td>C</td>
<td>TIME 5</td>
<td>CATMAINT UPDATE UNLDDN V12R1M505</td>
</tr>
<tr>
<td>V12R1M505</td>
<td>V12R1M500*</td>
<td>V12R1M505</td>
<td>V12R1M505</td>
<td>F</td>
<td>TIME 6</td>
<td>ACTIVATE FUNCTION LEVEL(V12R1M505)</td>
</tr>
</tbody>
</table>
New Catalog Columns

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Column Name</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSACCELERATEDPACKAGES</td>
<td>FNLEVEL</td>
<td>VARCHAR(10)</td>
<td>Function level of the package/query at the time the row was inserted</td>
</tr>
<tr>
<td>SYSDYNQRY</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSPACKCOPY</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSQUERY</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSCONTROLS</td>
<td>REGENERATETS</td>
<td>TIMESTAMP(12)</td>
<td>Time when the object was regenerated or altered</td>
</tr>
<tr>
<td>SYSINDEXES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSROUTINES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSTABLES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSKEYS</td>
<td>CREATEDTS</td>
<td>TIMESTAMP(12)</td>
<td>Time when the row was inserted</td>
</tr>
<tr>
<td>SYSENVIRONMENT</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Listed above are the new columns which have been added in places where we do not already have timestamp columns. We are not listing all the existing timestamp columns that could be used with SYSLEVELUPDATES in a similar fashion to that described on the previous slide.
Query to find the catalog level and function level when table MY.TABLE1 was created.

<table>
<thead>
<tr>
<th>FUNCTION_LVL</th>
<th>CATALOG_LVL</th>
<th>EFFECTIVE_TIME</th>
<th>OPERATION_TEXT</th>
</tr>
</thead>
<tbody>
<tr>
<td>V12R1M503</td>
<td>V12R1M500</td>
<td>TIME 3</td>
<td>ACTIVATE FUNCTION LEVEL (V12R1M503)</td>
</tr>
</tbody>
</table>

Query to find when function level 505 was activated.

<table>
<thead>
<tr>
<th>FUNCTION_LVL</th>
<th>PREV_FUNC_LVL</th>
<th>HIGH_FUNC_LVL</th>
<th>CATALOG_LVL</th>
<th>EFFECTIVE_TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>V12R1M505</td>
<td>V12R1M500*</td>
<td>V12R1M505</td>
<td>V12R1M505</td>
<td>2016-04-20-12.38.03.706031000000</td>
</tr>
</tbody>
</table>
Questions? ? ?
Summary

- Starting with DB2 12 New Function will ship in the maintenance stream (ML)
- Enabling new function is controlled by use of
  - Catmaint (CL)
  - -ACTIVATE FUNCTION LEVEL (FL)
  - APPLCOMPAT (APPLV)
- APPLCOMPAT is extended to apply to DDL, DCL, and to control IBM Data Server Driver Packages (ODBC, JDBC, .Net, PHP, RUBY... SAP).
- SYSLEVELUPDATES contains History on when system changes occurred
- Documentation will change to help highlight new function
- Working with Vendors to improve communication and early notification of changes, with access to PTFs prior to general availability
DB2 12 — The ultimate enterprise database for business-critical transactions and analytics

Thank you!
<table>
<thead>
<tr>
<th>DB2 for z/OS</th>
<th>DB2 for z/OS 12.0.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ Welcome to DB2 12 for z/OS</td>
<td></td>
</tr>
<tr>
<td>+ Getting started with DB2 for z/OS</td>
<td></td>
</tr>
<tr>
<td>+ What's new in the initial DB2 12 release?</td>
<td></td>
</tr>
</tbody>
</table>

---

### What's new in DB2 12 function levels?

- Available DB2 function levels
  - Function level 500 (enabled by migration to DB2 12)
  - Function level 100 (activated by migration to DB2 12)

- Summary of operational requisite changes for DB2 function levels
  - Code levels, catalog levels, function levels, and application compatibility
  - Activating new capabilities in DB2 12 function levels

---

**Info about the DB2 12 product at GA** (traditional “What’s New” information)

**Info about the DB2 12 function levels** (shipped by continuous delivery)

**Topics about the available function levels and their enhancements, with the most recent function level on top**

**Conceptual information about continuous delivery and instructions for activating new capabilities**
Tip: Bookmark this topic because it will be updated every time a new DB2 function level is shipped!

This topic has links to topics about each new function level, with the most recent on top.

The paragraph under the link summarizes what is included in the function level

Available DB2 function levels

DB2® for z/OS® delivers new capabilities as they become available, which means you can benefit from new capabilities without waiting for an entire new DB2 for z/OS release. After the code to support a particular enhancement has shipped in the service stream, you activate a function level to make the enhancements available in your DB2 environment.

Function levels are specified by 9-byte strings that correspond to the DB2 version, release, and maintenance value.

The format is VvvRrMmmm, where vv is the version, rr release, and mmmm is the modification level.

Sometimes a function levels are abbreviated. For example, “function level 501” refers to V12R1M501, which is the highest available function level in DB2 12.

Tip: To determine the catalog level and function level information for a DB2 data sharing group or subsystem and code levels of individual group members, issue a DISPLAY GROUP command. For examples of issuing this command, see DISPLAY GROUP (DB2). The DSN7100I message indicates the catalog, function levels of the data sharing group or subsystem, and the code levels of each data sharing member or subsystem. The DB2 LVL column indicates the code level of each data sharing member or subsystem. For detailed descriptions and example output, see DSN7100I.

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

**Function level 500 (enabled by migration to DB2 12)**
Function level function level 500 (V12R1M500) represents the first opportunity for applications to take advantage of most new function in DB2 12, including new SQL capabilities and subsystem parameter settings. In many respects, it is comparable to new-function mode in DB2 11.

**Function level 100 (activated by migration to DB2 12)**
In function level 100 (V12R1M100), DB2 runs on DB2 12 code. However, most new function, including new SQL capabilities and subsystem parameters are not enabled. In many respects, this function level is comparable to conversion mode in DB2 11.
Info about Individual Function Levels

The opening paragraph summarizes the enhancements.

Title of the function level nnn topics identifies the function level (V12R1Mnnn) and the APAR/PTF that enables it.

A table provides basic facts about each function level.
Information about Individual Function Levels (continued)

After the table of facts, you'll find one section for each enhancement in the function level, with a brief overview of that enhancement.

The overview will focus on **Who, What, and WOW!**

**Bitemporal support for DB2 Analytics Accelerator (V12R1M505)**

The DB2 Analytics Accelerator now supports both system-period and application-period temporal tables.

Bitemporal tables are ideal for applications that rely on tables that keep application-period information and system-based historical information because this type of table gives you a great deal of flexibility in how you query data based on periods of time. This DB2 Analytic Accelerator support of bitemporal tables improves the integration and power of the combination of DB2 for z/OS and the accelerator.

**Related information:**

- **Types of tables**
  - **ACCELERATION OPTIONS** field (QUERY ACCEL_OPTIONS subsystem parameter)

Each section for each enhancement concludes with links to:
- KC topics that provide more details about the enhancement
- Other resources (white papers, webcasts, etc.) about the enhancement
Details about Enhancements

ACCELERATION OPTIONS field (QUERY_ACCEL_OPTIONS subsystem parameter)

The QUERY_ACCEL_OPTIONS subsystem parameter controls whether certain types of queries are allowed to execute on an accelerator server. If you choose NO, you proceed to panel DSNTP8A after completing panel DSNTP02. If you choose YES, you proceed to panel DSNTP8A after completing panel DSNTP02.

Acceptable values: NONE, YES
Default: NONE
DSNZPxxx: DSN6SPRM_QUERY_ACCEL_OPTIONS (option NONE)

Parent topic: DSNTP02, Query accelerator preferences panel
Previous topic: GET ACCEL ARCHIVE field (GET ACCEL_ARCHIVE subsystem parameter)
Next topic: CURRENT QUERY ACCEL (QUERY_ACCELERATION subsystem parameter)

Related information:
Bitemporal support for DB2 Analytics Accelerator (V12R1M505)
Vendor Continuous Delivery Process
Pre-Continuous Delivery Vendor Process

- **Version upgrades**
  - Separate vendor program - early and regular vendor involvement and communication
  - Code drops starting over a year prior to ESP start
  - PFS roll-ups starting over a year prior to ESP start
- **Maintenance updates**
  - Notification of APARs affecting modules or control blocks that vendors have indicated they are sensitive to
  - Regular sharing of documentation for design changes delivered in maintenance stream
- **Improved communications to answer ad-hoc queries and aid resolution of high impact problems**
Vendor Continuous Delivery Support Summary

- Overriding aim is ensure stability of DB2 for z/OS environments
- Exploitation is desirable, but secondary concern
- Help achieve this through
  - 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 vendors and IBM development
- In spite of these efforts, Continuous Delivery places greater responsibility upon vendors to invest in ensuring stability and preventing regression from DB2 maintenance