



# How to Have a Happy DB2 11 CM Day

By Julian Stuhler, Triton Consulting

## Introduction

A lot has been said, written and presented regarding the many benefits and enhancements offered by DB2 11 for z/OS. Many DB2 sites are in the process of building cost cases to get management approval for the upgrade, or working with the business to agree implementation dates to minimise application impact. Other customers are deliberately building a delay into their implementation plans in order to allow the DB2 11 codebase to further mature before committing themselves<sup>1</sup>. Whatever the reason, many of you are looking at dates for the initial upgrade to DB2 11 Conversion Mode (CM) that are many months, perhaps even years, away. That delay can sometimes be frustrating from a technical perspective, but there is a significant amount of preparation work you can be doing well in advance of “CM Day” for any given DB2 environment.

This white paper outlines the major opportunities for preparation in advance of the move to DB2 11 Conversion Mode. Some of these changes require a full or partial outage, but if you are able to implement all or most of them as part of your routine maintenance activities, you will greatly reduce the risk and the elapsed time of the actual upgrade process.

Please note that this article relates specifically to preparing for an upgrade to DB2 11 for z/OS. Similar guidance on moving to DB2 10 for z/OS can be found in the previous version, which is still available from the [Triton Consulting Downloads](#) page.

In case you didn't pick up on this last time around, it's also worth noting that since the DB2 version number hit double-digits, IBM has had to move to a hexadecimal notation for the DB2 version/release code. So DB2 10.1 for z/OS was referred to “A10” in APAR text (and DSN10 for the standard library prefixes), and DB2 11.1 is therefore known as “B10”.

---

<sup>1</sup> Although it's worth noting that IBM put an increased emphasis on early production readiness during product development, and at least one customer went into production with DB2 11 for z/OS prior to GA. Feedback from ESP customers consistently praised the overall code quality and robustness of the new release.

## DB2 11 Advance Preparation Tasks

The following sections outline the major DB2 upgrade preparation tasks that you can undertake in advance, and are presented in no particular order.

### Get to DB2 10 New Function Mode

This might be stating the obvious, but you need to be in DB2 10 New Function Mode (NFM) before you begin your DB2 11 upgrade. For various reasons many sites tend to stay in Conversion Mode (CM) for extended periods – if you're one of them you need to ensure you get to New Function Mode (NFM) and are completely stable before even considering the next step in your DB2 11 journey.

Note that depending on your operational procedures and appetite for risk, an outage may be required to move to DB2 10 NFM.

### Check Your Hardware and Software Pre-Requisites

A full list of the hardware and software pre-reqs are listed in the DB2 11 for z/OS Program Directory, which can be found [here](#). For reference, the major items are:

- Hardware
  - z10 or later
  - zEC12 required for use of pageable 1MB page frames and 2GB page frames feature
  - Flash Express required for use of pageable 1MB page frames feature
  - Sufficient real storage for DB2 11 (see section on Check the Informational APARs and Stay **Up-To-Date on Maintenance**)

Each release of DB2 has one or more “informational APARs” dedicated to it, which provide a very useful single point of reference for important migration related maintenance. The DB2 11 APARs are [II14660](#) (for DB2 migration/fallback) and [II14732](#) (for DDF issues). [II14730](#) is also useful, as it covers DB2 Connect / IBM Data Server Client issues.

These APARs should be checked early and often to ensure that all of the recommended maintenance is applied to your DB2 environments prior to the move to Conversion Mode.

- **Check Real and Virtual Storage Availability** below)
- Firmware
  - CFLEVEL 17 required for use of new group bufferpool write-around protocol in a data sharing environment
- Operating System
  - z/OS 1.13 or later
  - APAR OA40967 required for use of 2GB page frames under z/OS V1.13

- APARs OA40966 and OA37550 required for use of new group bufferpool write-around protocol in a data sharing environment
- Associated Software – DB2 Connect / IBM Data Server driver
  - Any in-support release as a minimum
  - DB2 Connect V10.1 Fixpack 2 or DB2 Connect V9.7 Fixpack 6 (or later) recommended minimum for customers considering online migration
  - V10.5 Fixpak 2 is required for some new DB2 11 features such as array support, longer client information fields and improved autocommit performance)

## Check the Informational APARs and Stay Up-To-Date on Maintenance

Each release of DB2 has one or more “informational APARs” dedicated to it, which provide a very useful single point of reference for important migration related maintenance. The DB2 11 APARs are [II14660](#) (for DB2 migration/fallback) and [II14732](#) (for DDF issues). [II14730](#) is also useful, as it covers DB2 Connect / IBM Data Server Client issues.

These APARs should be checked early and often to ensure that all of the recommended maintenance is applied to your DB2 environments prior to the move to Conversion Mode.

## Check Real and Virtual Storage Availability

DB2 10 for z/OS eliminated the most pressing virtual storage restrictions by moving the majority of the remaining thread-related virtual storage areas above the 2GB bar. DB2 11 continues to exploit 64-bit virtual storage, and now reserves 1TB of contiguous 64-bit shared private storage for each DB2 subsystem on the LPAR (up from 128GB per subsystem in DB2 10). The HVSHARE z/OS parameter specifies the size of the high virtual shared area – this is set to a generous 510TB by default but it’s worth checking that this hasn’t been reduced at your site for some reason.

At the time of writing relatively little information is available regarding increased real storage requirements for DB2 11, although any increases are likely to be significantly lower than those required when upgrading to DB2 10. Early IBM testing indicated real storage increases in the 0-5% range, but allowing for an increase of up to 15% compared to DB2 10 real storage usage is a good idea.

If you have sufficient capacity within your current configuration you merely need to speak to your systems programmers/capacity planners to reserve some additional real storage for your future use. Otherwise, you may need to wait until the next scheduled hardware upgrade. Either way, you need to plan for the potential increase and be aware of the implications.

## Apply the DB2 11 Fallback SPE

The DB2 11 fallback SPE ([PM31841](#)) must be applied to **all** members of the data sharing group prior to any attempt to move to DB2 11 CM. The SPE can be applied one member at a time to maintain availability within a data sharing group, and before you move to DB2 11 CM each system needs to have been stopped and restarted at least once since the SPE was applied. The DSNHDECP module for each member also needs to be re-prepared, as documented in the SPE hold data.

As applying the SPE requires DB2 members to be stopped, you should plan to piggyback on a suitable change slot well in advance of your planned CM date.

## Run Pre-Migration Jobs

Most of the specific checks and potential inconsistencies listed throughout this section are covered by the DB2 11 pre-migration checker job, known as DSNTIJPM. This runs a series of reports highlighting potential issues within your current environment, and is an essential way of ensuring your system is in good shape prior to any migration attempt.

However, you don't need to wait until you have your DB2 11 libraries to run this job, as it's also possible to access it via APAR [PM94057](#) which provides an identical job within your current DB2 10 environment (except that it's then called DSNTIJPB).

Run DSNTIJPB against your DB2 10 systems early and often during your upgrade planning and preparation – you should aim to have a completely clean bill of health before attempting to go to CM (or at the very least be aware of any exceptions it is still reporting and have plans to address them).

## Convert Any Remaining Simple Tablespaces

DB2 9 withdrew support for the creation of simple tablespaces, although it's still possible to access existing simple tablespaces under DB2 9, DB2 10 and DB2 11. If you haven't already done so, you should put in place a firm plan for converting any remaining simple tablespaces to either segmented format or one of the new UTS formats. DB2 10 introduced some additional ALTER capabilities to make this job easier, so if you didn't address this issue when upgrading to DB2 9 or DB2 10 then now is the time!

Note that while the RECOVER utility still supports recovery of simple tablespaces in DB2 9 and beyond, this won't protect you from an accidental DROP TABLESPACE. If you're unable to convert all simple tablespaces in advance of the upgrade (which is definitely the recommended approach), you may want to consider issuing an ALTER TABLE ... ADD RESTRICT ON DROP statement against simple tablespace tables to reduce the chance of accidental data loss.

## Create a DB2 Performance Baseline

How well do you know the performance of your current critical transactions and batch processes? If there are any performance-related complaints or issues following the upgrade, how will you be able to really tell if there has been any regression?

Prior to any upgrade you should ensure you have a good picture of your current performance. Most DB2 performance monitors allow you to create and populate a “performance warehouse” that stores both detailed and summarised information about the performance of key processes. This data is critical to establish a “before” position for later comparison, but it’s surprising how many large, well-established DB2 sites don’t do this. If your site falls into this category, spend some time now to begin collecting the performance data so that you have a historical baseline to compare to after your entry to CM.

Don’t forget that most releases of DB2 involve changes to CPU accounting, with charging for some specific processes moving between the DB2 address spaces and allied address spaces. It’s therefore useful to collect historical data for both the system address spaces (via the Statistics trace) and allied threads (via Accounting) and consider the whole picture when comparing a given workload before and after the upgrade.

## Clean Up Your ZPARMs

How many DB2 systems programmers can place their hand on their heart and swear that the input members for the DB2 install CLIST are all up to date for all of their DB2 subsystems? It is common practice to directly update the DSNTIJUZ install job for DSNZPARM updates, which can cause problems with critical settings being regressed when the install CLIST is executed during upgrades and picks up old parameter settings.

For several releases now, IBM has provided a handy utility to generate a new DSNTIDxx install CLIST input member based on your current DSNZPARM and bufferpool settings. Job DSNTIJXZ allows you to create a fresh CLIST input member incorporating all of the current settings, ready for you to use when you run the install CLIST in order to move to DB2 11 Conversion Mode.

## Health Check Your Catalog & Directory

When was the last time you performed a hygiene check on your DB2 catalog and directory? A number of utilities exist to check these critical pagesets for internal consistency, and if you’re one of the many sites that don’t run them on a regular basis you should certainly do so before attempting a version upgrade. These include the DSN1COPY stand-alone utility with the PARM=CHECK option<sup>2</sup>, and the online CHECK DATA, CHECK INDEX and CHECK LOB utilities. If you can afford the outage, a reorg of

---

<sup>2</sup> Note that DSN1COPY may issue DSN1985I ZERO PAGES ENCOUNTERED. FIRST PAGE = first-zero-page-number, LAST PAGE = last-zero-page-number. This can be ignored on SYSSTR, SYSSTATS, SYSDDF, SYSUTILX and SPT01

the catalog is also worthwhile in order to speed up the catalog migration steps you're going to run later when moving to CM and NFM. Note that the DSN1CHKR utility has been deprecated in DB2 11, as it's no longer required now that the old internal links and hashes have been removed.

The utilities above should warn of any physical inconsistencies, but to check the logical consistency between catalog tables IBM also supplies a set of SQL statements in the shape of the DSNTESQ IVP job. This might take some time to run (depending on the size of your catalog) but it's worth the effort.

## Positive People Prepare Packages Pre-emptively

In an effort to reduce the overheads associated with supporting an ever-larger array of obsolete package structures, IBM has been getting stricter in terms of the oldest packages supported by recent DB2 releases. DB2 10 insisted that any packages bound earlier than Version 6 had to be rebound, and DB2 11 moves the goalposts again and finally achieves IBM's target of only supporting package structures from the two previous releases. Therefore, packages bound prior to Version 9 are not supported in DB2 11 upon entry to Conversion Mode<sup>3</sup>.

Part of your DB2 11 preparation should include identifying any pre-Version 9 packages and rebinding them in advance to avoid any nasty surprises if you allow DB2 to do it for you on CM day. Note that the DSNTIJPB job includes a report to help you identify these (see section on Pre-Migration Jobs above).

Don't forget that you have some powerful new facilities introduced in DB2 9 and 10 to help combat the possibility of access path regression during a REBIND. The "plan stability" feature allows you to keep either one or two previous copies of a package during the REBIND process. If you encounter any issues with the new package, simply use the SWITCH option to revert to the old copy.

The new APREUSE and APCOMPARE BIND options introduced in DB2 10 (and enhanced again in DB2 11) provide even more powerful options for minimising the possibility of access path regression during mass REBINDs. Unfortunately APREUSE and APCOMPARE need some new package structures that weren't introduced until DB2 9 so they are not going to help with these pre-V9 packages. These options will be of great value during your REBINDs once in DB2 11 CM though (and in preparing for future releases beyond DB2 11).

---

<sup>3</sup> DSNTIJPB/ DSNTIJPB will list all packages bound prior to DB2 9. Those still in existence on entry to CM will be dynamically rebound at runtime if you specify YES or COEXIST for the ABIND DSNZPARM. Otherwise any attempt to execute them will fail with SQLCODE -908, SQLSTATE 23510 (for packages) or SQLCODE -923, SQLSTATE 57015 (for plans).

## Check EXPLAIN Table Format Validity

DB2 11 makes changes to the EXPLAIN tables and you should plan to upgrade all of your tables to the V11 format once you're in CM. DB2 11 will tolerate V9 and V10 format tables (with a warning message being issued), but EXPLAIN won't run against pre-V9 format tables. If you have any pre-V9 specimens lying around consider updating them to V10 format before moving to DB2 11 CM (or plan take them direct to V11 format immediately after CM).

## Check for use of Unsupported Stored Procedures and Functions

SYSPROC.DSNAEXP is an IBM-supplied sample stored procedure that enabled you to execute an EXPLAIN statement for certain simple SQL statements without having the authorization to execute that SQL statement. This procedure was deprecated in DB2 10 (replaced by the new SQLADAM authority and EXPLAIN privilege) and support has been removed completely in DB2 11.

Similarly, the DB2 MQ functions that used the Application Messaging Interface (AMI) were deprecated back in DB2 9, and will no longer work from DB2 11 onwards. You should already have converted all AMI-based MQ function calls to the newer MQI equivalents, but if you haven't you'll need to do so before moving to DB2 11 CM. As the MQI versions are mostly direct replacements for their AMI counterparts with the same function signatures, this should be a relatively straightforward process.

DSNTIJPM/DSNTIJPB will report on any instances of SYSPROC.DSNAEXP or AMI-based MQ functions.

## Remove Views, MQTs, and SQL table functions with period specifications

Temporal support was only introduced in DB2 10, so it's somewhat unusual to have to check for incompatibility with such a new feature. However due to some changes in the way temporal support is implemented in DB2 11, views, MQTs, and SQL table functions that contain a SYSTEM\_TIME or BUSINESS\_TIME period specification might return incorrect results in DB2 11 CM, and should be dropped. DSNTIJPM/DSNTIJPB contains some catalog queries to help identify these.

## REORG TABLESPACE SHRLEVEL NONE on LOB Tablespaces

DB2 10 implemented much better support for reorganising LOB tablespaces, and in the process REORG SHRLEVEL NONE support for LOB tablespaces was deprecated. Once in DB2 10 NFM, a REORG SHRLEVEL NONE utility executed on a LOB tablespace will issue a DSNU126I message to indicate that the function is no longer supported, but strangely the step will complete with RC=0 even though nothing has actually been done to the tablespace you thought you were reorganising.



Once you're in DB2 11 CM the DSNU126I message will still be issued, but the step will complete with RC=8. This is better than the DB2 10 behaviour in that you now know you're doing nothing in that job step, but it's worth checking for these and changing them to SHRLEVEL REFERENCE or SHRLEVEL CHANGE before you move to CM in order to avoid unnecessary job failures and/or overnight call-outs.

## **10-Byte RBA and LRSN Values**

DB2 11 expands RBA and LRSN values from 6 to 10 bytes in order to dramatically increase the addressable log range. Although specific steps are required in NFM to convert your BSDS and system/application pagesets to the new 10-byte formats, DB2 11 will still use 10-byte addressing internally from the moment you first start it in CM.

This means that `-DISPLAY THREAD` commands will show a 10-byte URID value (the 6-byte value left-padded with zeroes) from CM onwards. If you have any processes or utilities that parse URID values, these must be prepared to accommodate the new 10 byte values.

Similar considerations apply to processing of DB2 trace records that contain RBA/LRSN values, as these will also be expanded to 10 bytes upon entry to CM. In most cases these fields have been moved so other offsets in the record are not impacted, but there are some exceptions. The DB2 11 Installation and Migration Guide contains detailed information on how to change programs to account for increased sizes and new offsets.

## **Expanded DRDA Client Information**

DB2 11 expands the length of several of the DRDA client information fields, which are used to pass additional context information from DRDA clients to the DB2 server. Upon entry to DB2 11 CM, any messages that display this client information (e.g. `-DISPLAY THREAD`) will use the new message lengths. If you have any processes that parse the output from these commands, they must be prepared to accommodate the expanded field lengths.

## **Password protection of Active / Archive Logs**

Previous releases of DB2 allowed the active and archive log datasets to be password protected. This is no longer supported from DB2 11 onwards, so these passwords should be removed prior to moving to DB2 11 CM.

## **Application Incompatibilities**

DB2 11 introduces some changes that may require application changes, as new releases always do. You should research the full list documented in the DB2 Installation and Migration Guide, but the good news is that IBM is trying to ease the process of identifying

potential incompatibilities well in advance. [PM84769](#) is a DB2 10 APAR that extends IFCID366 to report on some of the new reserved words in DB2 11, so you can have plenty of advance warning on many of the changes needed to your the applications.

Of course, most application release incompatibilities don't actually need to be addressed until you move to DB2 11 New Function Mode (NFM), so if you haven't already resolved them beforehand you can use DB2 11's extended IFCID 366/376 trace records while you're in CM to help to identify any incompatible SQL/XML that may be executed. Finally, if some applications still haven't made the required changes when you want to move the environment to NFM you can use DB2 11's new APPLCOMPAT feature to allow specific packages continue to run in DB2 10 compatibility mode.

For those of you that are still unable or unwilling to get off those old, unsupported compiler releases, IBM have relented yet again and provided support for the old DB2 V7 pre-compiler (DSNHPC7).

## Check DB2 Tool Compatibility

Most sites rely heavily on additional tools and utilities to supplement the functionality within the base DB2 product. These range from performance monitors to change administration tools and data utilities. As part of your preparation, you will need to work with the tool vendor to ensure that each one at least tolerates (and ideally fully exploits) DB2 11.

This used to be a fairly straightforward issue, but recently some of the major tools vendors have been a little slower to fully exploit all of the functionality in new DB2 releases. As always, talk to your tools vendor(s) well in advance and plan accordingly. Pay particular attention to replication and recovery management products, which will require significant changes to handle the new 10-byte log addresses discussed earlier. This includes anything that uses IFCID 306 (such as IBM's own InfoSphere Data Replication and InfoSphere Change Data Capture products) as well as those that exploit a user-written log capture exit.

## Test, Test, Test

Even if your first "real" DB2 11 upgrade is still many months away, you can and should practice the entire upgrade/fallback process in a "sandpit" DB2 system, and then make it available to allow developers, DBAs and systems programmers to become familiar with the new functions and features. Cloning your production catalog/directory and practicing the migration process is also an excellent idea, and will give you a good indication of the elapsed time of the process when you do it for real.

If you're in a data sharing environment and intending to minimise application impact by upgrading one member at a time, your testing should include running workload in co-existence conditions with one or more members at each release to ensure results are as

expected. The DB2 Installation and Migration Guide includes some specific guidance on release co-existence (for example the new 64-bit Java Virtual Machine (JVM) for stored procedures cannot be used until all members are running under DB2 11 code).

It may be possible to order a “trial” version of DB2 11 to ensure that the clock doesn’t start ticking on your Single Version Charging (SVC) agreement, but you should work with your local IBM representative to see what is possible.

## **Organise DB2 11 Education**

Take advantage of the time leading up the upgrade to get yourself and your colleagues fully educated on the many new facilities and features within DB2 11. Some items such as the new Application Compatibility (APPLCOMPAT) feature could make life much easier when the time comes to move to DB2 11 New Function Mode.

There is a wealth of technical information available on the Internet and at various DB2 conferences such as IDUG to provide this information. In addition, IBM offer free Migration Planning Workshops and third-parties such as Triton Consulting can provide courses or consultancy tailored to your specific requirements and issues.

## **Summary**

This white paper concentrated on the major tasks you can undertake in advance of your actual move to DB2 11 Conversion Mode. If you are able to complete all or most of these you can greatly reduce the elapsed time and risk associated with the actual upgrade, and make your “DB2 11 CM day” a happy one.

For a full list of the pre-requisites for DB2 11, please refer to the DB2 11 for z/OS Installation and Migration Guide. A handy pre-migration checklist is also available online – [see here](#).

Good luck with your upgrade!