

WHITE PAPER | OCTOBER 2014

# Best Practices for Optimizing Your Mainframe and Upgrading to DB2 10 for z/OS

Troy Coleman  
CA Technologies Product Marketing



## Table of Contents

---

<b>Executive Summary</b>	<b>3</b>
<b>Section 1:</b> Preparing to Upgrade	<b>4</b>
<b>Section 2:</b> Dealing with Deprecated Features	<b>5</b>
<b>Section 3:</b> Skip-Level Migration	<b>7</b>
<b>Section 4:</b> Rebinding Programs and Managing Access Paths	<b>8</b>
<b>Section 5:</b> Introducing New Features	<b>9</b>
<b>Section 6:</b> Performance Management	<b>10</b>
<b>Section 7:</b> Conclusion	<b>11</b>
<b>Section 8:</b> About the Author	<b>12</b>

## Executive Summary

---

### Challenge

Every three years or so each DB2 shop is tasked with upgrading its DB2 for z/OS software. Maintaining currency with DB2, at least inasmuch as ensuring that your version of DB2 software is still supported by IBM, is a non-negotiable part of running mainframe database management software.

---

### Building a Plan

You should plan on upgrading to a new version of DB2 approximately every three years. To understand why, look at the table below, which offers up the recent history of DB2 releases.

DB2 Version	General Availability	End of Marketing	End of Service
5	1997-06-27	2001-12-31	2002-12-31
6	1999-06-15	2002-06-30	2005-06-30
7	2001-03-30	2007-03-05	2008-03-30
8	2004-03-26	2009-09-08	2012-04-30
9	2007-03-06	2012-12-10	2014-06-27
10	2010-10-22	-	-

As you can see, at any point in time there are several versions of DB2 that are current—that is, they are supported by IBM. If you have problems with DB2, IBM will provide technical support and help you to resolve those issues. However, after a period of time, versions are no longer supported, as indicated by the end of service date. (The end of marketing date is the date by when IBM will no longer sell that version of the software.)

So, as of summer 2013, the current version is DB2 10 for z/OS, with DB2 9 also supported. DB2 version 8 is no longer supported as of April 30, 2012. Also looming is DB2 11 for z/OS, which was made available to a select group of clients in the Early Support Program (ESP) on March 8, 2013. DB2 11 for z/OS will likely become generally available toward the end of 2013.

By analyzing the recent history of DB2 for z/OS, we can easily determine that new versions are released by IBM approximately every three years, and that once IBM stops marketing a version of DB2, users will have between one and three years before end of service.

It is vitally important for organizations to ensure that they are using only supported versions of DB2 for z/OS. Therefore, it is essential to prepare an upgrade strategy for your organization. Doing so requires that you be able to answer the following questions:

- Do you wish to be bleeding edge and upgrade to the latest version of DB2 as soon as it becomes available?
- If not, how long will you wait before upgrading?
- Or is your organization a late adopter? Will you plan to stay one version behind current (as long as it is still supported)?

Once you have answered these questions, you can embark on building a plan for DB2 upgrades. But understand that there is no option to avoid upgrading. As such, every shop needs to understand the primary issues discussed in the ensuing sections of this technology brief. Then plan accordingly based on how up to date you want your DB2 environment to be.

---

## Section 1:

### Preparing to Upgrade

The first task for any DB2 upgrade plan is to understand the prerequisites for the new version you're planning to install. "Prerequisites" refers to the hardware and software that is required in order to run DB2.

IBM will provide a list of prerequisites for each new version. These will list the minimum version of the z/OS operating system and the oldest machine and processor that can still be used to run that version of DB2.

For example, for DB2 11 for z/OS, the key prerequisites are:

- z/OS V1.13 Base Services (5694-A01) or later
- DB2 11 for z/OS operates on z10TM or higher, processors supported by z/OS V1.13 or later

New reserved words in a DB2 version can cause additional problems if they are not uncovered and dealt with during the upgrade. A reserved word is a term that is used by DB2 and therefore cannot be used in certain contexts because DB2 may interpret it as a SQL keyword. Failing to identify whether or not the new reserved words are used in your programs can cause applications to fail post-upgrade.

For DB2 10 for z/OS, there are three new reserved words:

- Period
- Organization
- Zone

Reserved words are documented in the *IBM DB2 SQL Reference Guide* manual. IBM SQL has additional reserved words that DB2 for z/OS does not enforce but that nevertheless should not be used. These reserved words are documented in the *IBM DB2 SQL Reference for Cross-Platform Development* manual.

Of course, there will be additional prerequisites that you will need to investigate. For example, do you have sufficient storage and memory required for the new version? Additionally, you must make sure that required maintenance has been applied to the existing installation of DB2 and perhaps z/OS.

It is also imperative that you prepare your system for fallback before you attempt to upgrade. Fallback refers to going back to the current version of DB2 at some point in the future after you have started to upgrade to the new version. You may need to fallback for any number of reasons, including deprecated features (discussed later in this brief) or performance. Preparing for fallback usually involves applying PTF maintenance to the DB2 subsystem before beginning the upgrade process.

*The IBM DB2 Installation and Migration Guide* manual, which is available for every version of DB2 (for DB2 10 for z/OS the manual number is GC19-2974), can guide you through all the prerequisites.

---

## Section 2:

# Dealing with Deprecated Features

## What is deprecation?

A deprecated feature or function is one that has been outdated by newer constructs. Deprecation implies eventual obsolescence, but for the version of DB2 in question deprecated features are still supported.

It is important to begin avoiding the use of all deprecated features and functions to minimize your workload. Continuing to implement new systems that rely on deprecated functionality will require more conversion work when the feature is eventually dropped from support.

A prime example of a deprecated feature is support for simple table spaces, which was deprecated in DB2 9 for z/OS. For DB2 9 and later versions, it is no longer possible to create new simple table spaces, but your existing simple table spaces will continue to operate as before. Additional functionality is deprecated in DB2 10 for z/OS, including:

- **Several DB2-supplied stored procedures**, including SYSPROC.DSNACCOR (replaced by SYSPROC.DSNACCOX), SYSPROC.DSNAEXP (replaced by the EXPLAIN privilege), SYSPROC.DSNTBIND (replaced by SYSPROC.ADMIN\_COMMAND\_DSN), SYSPROC.DSNUTILS (replaced by SYSPROC.DSNUTILU) and SYSPROC.DSNWZP (replaced by SYSPROC.ADMIN\_INFO\_SYSPARM).
- **Several DB2-supplied user-defined functions**, including DB2XML.SOAPHTTPC, DB2XML.SOAPHTTPV and SYSFUN.DSN\_XMLVALIDATE (replaced by the DSN\_XMLVALIDATE built-in function).
- **Several DSNZPARMs**, including ASSIST, COORDNTR, DISABSCL, DPSEGSZ, OJPERFEH, OPTIOPIN, OPTIOWGT, OPTIXIO, PTCDIO, RETVLCFK, SEQCACH, SEQPRES, SMSDCFL, SMSDCIX and STATCLUS.
- **NEWFUN(YES) and NEWFUN(NO)**, replaced by NEWFUN(V8), NEWFUN(V9) and NEWFUN(V10).
- **Other features** including Sysplex query parallelism, DSN1CHKR utility, EXPLAIN tables in earlier formats, DDF\_COMPATIBILITY(SP\_PARMS\_NJV), password protection for active and archive log data sets, and ENABLE/DISABLE on BIND PACKAGE to enable or disable specific remote connections.

## Obsolete features

Even more important than identifying deprecated features is identifying and planning your immediate migration away from features that are made obsolete in the new version. An obsolete feature will simply no longer work. This means that you must remove the feature or convert your database or application so that it no longer uses that feature.

A prime example of an obsolete feature is plans with DBRMs bound directly into them, which has been rendered obsolete as of DB2 10 for z/OS. Moving to DB2 10 requires the conversion of all DBRMs into packages before they are bound into plans.

It is important to understand the distinction:

- Deprecated features are earmarked for obsolescence but can still be used, for the time being.
- Obsolete features are no longer supported and must be removed or your applications will fail.

You can find the deprecated and obsolete features for a new version in the IBM DB2 Installation and Migration Guide manual. In addition, recent versions of DB2, beginning with DB2 V6, have provided DSNTIIPM, an automated routine that can assist in determining the readiness of your DB2 subsystems prior to upgrading. Running DSNTIIPM on the prior release will identify conditions that can impede migration to the new release. By examining the generated report and correcting the issues it uncovers, you can assure the readiness of your DB2 subsystems for new versions.

## Converting away from deprecated and obsolete features

Of course, once you have identified the obsolete and deprecated features in use by your DB2 subsystems, the next step is to decide what to do.

Deprecated features need not be treated as emergencies and do not need to be immediately remediated. Remember, features that have been deprecated are still supported but will be removed in a future version of DB2. It's a good idea to examine the list of deprecated features and determine the effort involved in removing or converting them. But it is probably not a good idea to work on deprecated features as part of an upgrade. You can always remove or convert the deprecated features at a later date, after you have successfully upgraded.

Obsolete features, of course, need to be removed or migrated to use other supported functionality.

## How CA Technologies can help remediate obsolete features

CA Technologies offers solutions designed to automate DB2 database change management to reduce the amount of time, effort and human error involved in eliminating obsolete functionality.

**CA RC/Update™ for DB2 for z/OS** automates the tasks involved in changing DB2 objects and data.

It provides a development environment for the application developer, an editor and data copy feature for the end user, and sophisticated object management facilities for the DBA. After you have identified the obsolete features to change, you can deploy CA RC/Update to make the appropriate changes.

**CA RC/Migrator™ for DB2 for z/OS** automates the migration of changes from one DB2 environment to another, such as from test to production. So you can implement your changes to remove obsolete features using CA RC/Update and then migrate those changes to your other subsystems using CA RC/Migrator.

The combination of CA RC/Update and CA RC/Migrator can minimize the effort involved in remediating obsolete features during a DB2 upgrade, resulting in reduced downtime and increased DBA productivity.

---

### Section 3:

## Skip-Level Migration

Skip-level migration is the ability to upgrade to a new DB2 version without first having upgraded to the immediate past version. For example, it is possible to upgrade to DB2 10 for z/OS directly from DB2 Version 8 new function mode (NFM) without having to first upgrade to DB2 9 for z/OS.

Not every version of DB2 supports skip-level migration. To date, IBM has enabled skip-level migration twice, approximately every 10 years:

- To DB2 Version 7 from DB2 Version 5, skipping over DB2 Version 6
- To DB2 10 for z/OS from DB2 Version 8, skipping over DB2 9 for z/OS

DB2 11 for z/OS will not support skip-level migration.

Just because skip-level migration is offered does not mean that you must use it. The primary reason for skip-level migration is to enable IBM's customers to catch up to a more current version of DB2. According to estimates provided by IBM at IDUG 2013, approximately 1 customer in 5 migrated using a skip-level migration for V5 to V7, and the numbers are similar, so far, for DB2 10 for z/OS from DB2 V8.

When skip-level migration support is available, each organization will need to evaluate the pros and cons of skipping. The first issue that must be addressed is to gauge your organization's "bleeding edge" tolerance. If you are already two versions behind the current DB2 version, there is probably a reason. Perhaps your organization moves slower, letting others shake out the inevitable problems encountered in a new DB2 version. Or maybe you lack the manpower or budget to move quickly to new releases. If your organization falls into either of these categories, eschewing skip-level migration may be the best option.

Another factor to consider is that a skip-level migration project requires a longer timeline and involves more risk. Why is this so? Testing and rollout require more effort and planning than for a single version migration due to the additional features of two versions (e.g. DB2 9 and DB2 10) instead of one. And education and remediation work will be close to double the effort. A good project plan estimate is about 150 percent of a single release migration.

The final element to consider is timing. If your upgrade project is delayed too long, you could find your organization quickly facing the prospect of running an unsupported version of DB2. For example, any organization still on DB2 V8 after April 30, 2012 will be running on a version of DB2 that is past its end of service date.

Of course, the primary motivating factor for determining when you should upgrade to a new version of DB2 should be the need to utilize new features available only in the new version. This could be the desire to implement an application using temporal tables or simply to exploit new efficiencies for performance gains.

## Section 4:

# Rebinding Programs and Managing Access Paths

Managing access paths is probably the most time-consuming and potentially performance-damaging aspect of upgrading to a new version of DB2. Although it is not usually a hard and fast requirement to REBIND all your packages and plans when you move to a new version of DB2, it is a very good idea to do so, and it's typically recommended by the IBM installation guidelines.

Why is this so? First, there are a lot of optimizer enhancements and performance improvements that you won't experience without a REBIND. Only by rebinding your programs will you be able to use most of the new optimization features that come with the new version of DB2. Failing to do so means you're not taking advantage of the features you're paying for.

Second, keep in mind that there are some REBINDs that you simply cannot avoid. With each new version of DB2, IBM eliminates support for programs that were last bound on very old versions of DB2. For example, with DB2 10 for z/OS:

- Plans and packages from DB2 V5 or earlier will be automatically rebound when accessed by DB2 10.
- Plans bound with DBRMs are automatically rebound to convert them to a PKLIST, and the DBRMs are bound into packages.
- DB2 incrementally rebinds static SQL statements that use parallelism after migration to version 10.

It is a best practice to avoid having to REBIND a program when it is accessed, because doing so will cause a delay that will be experienced as a performance issue by the end user. The last thing you want is for your users to think a new version of DB2 is causing performance issues.

Other features of a new version may also compel you to perform additional rebinding. For example, DB2 10 for z/OS provides thread constraint relief, which improves the number of concurrent threads that can be running. To get this relief, though, you must REBIND, which moves work areas above the bar. This should result in the ability to run much more concurrent work in a single subsystem and to run from 5 to 10 times more concurrent threads.

The bottom line is that you will need to REBIND programs as part of upgrading to a new version of DB2. You may take the recommended approach of rebinding everything to garner the benefit of the new optimization functionality, or you may choose to rebind only those programs that must be rebound. Either way, you will need to invest time and energy in managing your DB2 access paths during a DB2 for z/OS upgrade.

## Access path degradation

Of course, the problem with rebinding is that, in some cases, access paths can get worse afterward. The possibility of degraded performance is real; the problem is finding the statements that may have worsened prior to experiencing an impact in production systems.

Before rebinding, it is imperative to ensure that the objects being accessed are properly organized and that appropriate RUNSTATS have been run. This gives DB2 the best chance for building efficient access paths when rebinding programs. Take care in determining when to execute the RUNSTATS

utility when rebinding for version upgrades. For example, IBM modified the algorithm for RUNSTATS in DB2 9 for z/OS, resulting in a different method of determining cluster ratio. So it can make a big difference when upgrading from DB2 V8 if you ran RUNSTATS before the upgrade as opposed to afterward.

DB2 10 for z/OS introduced several new bind/rebind options to help manage and control efficient access paths. The first parameter, named APRETAINDUP, can be used with either EXTENDED or BASIC plan management. When it is set to NO, old copies are not saved if the new access paths are identical to the old ones. Using this option is a good idea because the only reason to save a backup copy of access paths is if the new access paths are different.

Another parameter, APREUSE, can be used to implement access path reuse for both BIND and REBIND. Specifying YES causes DB2 to try reusing existing access paths. For a BIND, of course, DB2 can only try reusing access paths for queries that have not changed.

The final new parameter, APCOMPARE, enables access path comparison. This enables DB2 to alert you when access paths change during a BIND or REBIND.

All of these options are helpful, but they do not remove the need to manage and control your program REBIND process when upgrading to a new version of DB2.

## Resolving issues with CA Technologies solutions

CA Technologies offers solutions for managing the program REBIND process.

**CA Plan Analyzer® for DB2 for z/OS** can be used to analyze the SQL statements in your programs using expert rules to recommend techniques for improving performance. By integrating CA Plan Analyzer into the REBIND process during your DB2 upgrade, you can better identify any potential degraded access paths, along with suggested corrections to improve performance.

**CA Database Analyzer™ for DB2 for z/OS** provides in-depth statistics gathering functionality that can be used to extend and replace the statistics gathered by IBM RUNSTATS.

---

## Section 5:

### Introducing New Features

Part of your DB2 upgrade process should be the formation of a plan for introducing new functionality and features to your user base. This requires evaluating the new functionality within a new DB2 version, which necessitates some amount of education. There are multiple options for educating yourself on new versions. Some of the free options include:

- IBM manuals and RedBooks
- Webinars conducted by IBM, vendors and consultants
- Articles and blogs

Of course, the best option for learning about a new version of DB2 is to attend a training course, which will add to the cost of the upgrade. It's a good idea to send one DBA to the training course and then have that DBA train the rest of the organization. It is important, though, to make sure that training is available internally for DBAs, systems programmers, programmer/developers and end users.

With the proper education under your belt, you can then determine—on a feature-by-feature basis—which new functionality should be adopted by your organization and by which applications. For example, after you migrate to DB2 10 for z/OS, temporal support becomes available. But building temporal tables and applications is not a simple task and should only be attempted after gaining a firm understanding of the technology and its applicability to your particular business needs.

In most cases, new functionality is available only in new function mode. However, some enhancements work in conversion mode. It is important to differentiate between the two types of functionality and to understand and communicate which features require new function mode and which will be available immediately. Failure to do so can result in surprises (when new functionality appears earlier than expected) or disappointment (when new functionality does not appear).

At any rate, education and communication is an essential element of a successful DB2 upgrade. If you don't let your IT personnel know about the new features of the new DB2 version, one of two bad things will occur: either nobody will use the new functionality, or people will use it without the requisite training.

---

## Section 6:

### Performance Management

Perhaps the most important consideration during a version upgrade is the performance expectations for DB2 applications when running on the new version. There are several crucial aspects of performance that need to be understood and measured as part of the upgrade process.

The first aspect of performance management during an upgrade is the expected out-of-the-box (OOTB) improvement or degradation. OOTB implies that nothing else is done other than upgrading the DB2 system software: no rebinding, no ZPARM adjustments, etc. Understanding the expected OOTB performance delta will clarify the types of changes that you should consider making as part of the DB2 upgrade.

Although IBM strives for each new version of DB2 to improve OOTB performance, at times that goal is not met. In other words, OOTB performance of your DB2 application post-upgrade can be better than or worse than with the previous version.

New versions of DB2 can also come with recommended changes. Most of these changes can improve performance, but some can potentially degrade performance. Recommended changes are just that: not required but strongly suggested. A new version of DB2 can also come with required changes that must be implemented. Your job will be to ensure that all required changes are made, as many recommended changes as possible are made, and all with a positive impact on performance.

This can be a challenge to achieve. It requires patience, planning and timing. You should make sure that you have a baseline of performance information for your existing installation prior to upgrading. If you do not have such a baseline, it will be impossible to compare the new environment to the old.

You should also create additional performance baselines along the way. For example, it can make sense to create a baseline for:

- The upgraded DB2 environment immediately after the upgrade and with no changes implemented (if possible)
- The upgraded DB2 environment after making required changes
- The upgraded DB2 environment after making recommended changes

By doing so you will be armed with the performance measurements for all the variations of the system along the way, and you can make informed evaluations of the performance of your DB2 systems at each step in the upgrade process.

### How CA Technologies can help with performance

Tackling performance as part of an upgrade can be a significant task, especially without the help of automated solutions. CA Technologies offers several performance solutions that can be of great help during your upgrade.

**CA Detector® for DB2 for z/OS** can be used to evaluate application performance by monitoring information about your static and dynamic SQL statements for both real-time and historical performance analysis. **CA Insight™ Database Performance Monitor for DB2 for z/OS** is a real-time, online performance monitoring solution that can be used to examine your DB2 subsystems and DB2 applications, including network-connected applications outside of the z/OS environment.

One of the important components of your DB2 upgrade should be the analysis and verification of the performance characteristics of the new DB2 version vis-à-vis the previous version. With a performance warehouse based on CA Detector or CA Insight data, you can report average CPU usage before the upgrade and average CPU usage afterward.

---

#### Section 7:

## Conclusion

Upgrading your DB2 systems should be a regular, planned part of your database administration efforts, and DB2 tools from CA Technologies can reduce the cost and risk associated with upgrades. Approximately every three years there will be a new version of DB2, and if you do not plan for and adopt new releases on a regular basis you will be in danger of running unsupported software.

But with a plan, a little bit of rigor and the assistance of CA Database Management Utilities Suite for DB2 for z/OS, you can be sure of being up to date and ready for each new version of DB2.

**Section 8:**

## About the Author

Troy L. Coleman, principal product manager for CA Technologies, is responsible for CA Mainframe Chorus and DB2 Database Management. As an officer and webmaster for the Midwest Database User Group, he has extensive experience in all facets of database systems development, including systems analysis and design, database and system admin, data analysis and application development. Troy was the architect and lead developer of the first wireless database management software for DB2 UDB, PocketDBA, by Expand Beyond Corp. He is an internationally recognized consultant and lecturer, specializing in DB2 database administration and application architecture. Troy is a former adjunct professor teaching database management systems at Concordia University in River Forest, Illinois.



Connect with CA Technologies at [ca.com](http://ca.com)



CA Technologies (NASDAQ: CA) creates software that fuels transformation for companies and enables them to seize the opportunities of the application economy. Software is at the heart of every business, in every industry. From planning to development to management and security, CA is working with companies worldwide to change the way we live, transact and communicate – across mobile, private and public cloud, distributed and mainframe environments. Learn more at [ca.com](http://ca.com).