



Craig S. Mullins

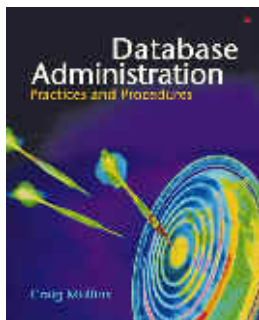
[Return to Home Page](#)

February 2008



The DBA Corner

by Craig S. Mullins



[Keeping Up With the DBMS](#)

One of the more troubling trends for DBAs is keeping up with the latest version of their DBMSs. Change is a fact of life and each of the major DBMS products change quite rapidly. A typical release cycle for DBMS software is 18 to 24 months for major releases with constant bug fixes and maintenance delivered in between major releases. Indeed, keeping DBMS software up-to-date can become a full-time job.

The troubling aspect of DBMS release migration these days is that increasingly, the majority of organizations are not on the most recent version or release of the software. The uptake of Oracle 11g, released earlier this year, is slow. Many mainframe organizations are still struggling with DB2 V8 and therefore very few have moved to DB2 9 for z/OS yet. This happens for many reasons including the desire to let others work out the inevitable early bugs, the lack of compelling new features that would drive the need to upgrade immediately, and lack of time to adequately upgrade as often as new releases are unleashed on us.

The DBA team must develop an approach to upgrading DBMS software that conforms to the needs of their organizations and minimizes the potential for disrupting business due to outages and database unavailability.

You may have noticed that I use the terms version and release somewhat interchangeably. That is fine for a broad discussion of DBMS upgrading, but a more precise definition is warranted. Versions typically are very broad in scope, with many changes and new features. A release is typically minor, with fewer changes and not as many new features. But DBAs must meticulously build implementation plans for both.

In many cases, upgrading to a new DBMS version can be treated as a special case of a new installation. All of the procedures required of a new installation apply to an upgrade: you must plan for appropriate resources, you need to reconsider all system parameters, and you need to ensure that all supporting software is appropriately connected. But there is another serious issue that must be planned for, and that is existing users and applications. An upgrade needs to be planned so as to cause as little disruption to the existing users as possible. Therefore, upgrading can be a tricky and difficult task.

Keeping the DBMS running and up-to-date without incurring significant application outages requires an on-going effort that will consume many DBA cycles. The approach undertaken must conform to the needs of their organization, while at the same time minimizing business impact and avoiding the need to change applications.

Upgrading to a new DBMS release offers both rewards and risks. By moving to a newer DBMS release developers will be able to use the new features and functionality delivered in the new release. For purchased applications, you need to be cognizant of the requirements of application releases on specific DBMS versions. Additionally, new DBMS releases tend to deliver enhanced performance and availability features that can optimize existing applications. Often the DBMS vendor will provide better support and respond to problems faster for a new release of their software. DBMS vendors are loath to allow bad publicity to creep into the press about bugs in a new and heavily promoted version of their products. Furthermore, over time, DBMS vendors will eliminate support for older versions and DBAs must be aware of the support timeline for all DBMSs they manage.

An effective DBMS upgrade strategy will balance the benefits against the risks of upgrading to arrive at the best timeline for migrating to a new DBMS version or release. An upgrade to the DBMS almost always involves some level of disruption to business operations. At a minimum, as the DBMS is being upgraded databases will not be available. This can result in downtime and lost business opportunities if the DBMS upgrade has to occur during normal business hours (or if there is no planned downtime). Other disruptions can occur including the possibility of having to convert database structures, the possibility that previously supported features were removed from the new release (thereby causing application errors), and delays to application implementation timelines.

The cost of an upgrade can be a significant barrier to DBMS release migration. First of all, the cost of the new version must be planned for (price increases for a new DBMS version can amount to as much as 10 to 25 percent). You also must factor in the costs of planning, installing, testing, and deploying not just the DBMS but also any applications using databases. Finally, be sure to include the cost of any new resources (memory, storage, additional CPUs, etc.) required by the DBMS to use the new features delivered by the new DBMS version.

Also, in many cases the performance benefits and improvements implemented in a new DBMS release requires the DBA or programmers to apply invasive changes. For example, if the new version increases the maximum size for a database object, the DBA may have to drop and re-create that object to take advantage of the new maximum.

Another potential risk is the possibility that supporting software products may lack immediate support for a new DBMS release. Supporting software includes the operating system, transaction processors, message queues, purchased application, DBA tools, development tools, and query and reporting software.

And we haven't even touched on applying maintenance to the DBMS. The bottom line is that keeping up with new DBMS releases and functionality has become a very significant component of the DBA's job.

From [Database Trends and Applications](#), February 2008.

© 2008 Craig S. Mullins, All rights reserved.

[Home](#).