

May 2010

The Essential Guide to **SQL Server 2008 R2 for the DBA**

By Allan Hirt



Microsoft®
SQL Server® 2008 R2

In May 2010, Microsoft is releasing SQL Server 2008 R2. More than just an update to SQL Server 2008, R2 is a brand new version. This essential guide will cover the major changes in SQL Server 2008 R2 that DBAs need to know about. For additional information about any SQL Server 2008 R2 features not covered, see <http://www.microsoft.com/sqlserver>.

Editions and Licensing

SQL Server 2008 R2 introduces two new editions: Datacenter and Parallel Data Warehouse. I will discuss Parallel Data Warehouse later, in the section “BI enhancements in SQL Server 2008 R2.”

SQL Server 2008 R2 Datacenter builds on the value delivered in Enterprise and is designed for those who need the ultimate flexibility for deployments, as well as advanced scalability for things such as virtualization and large applications. For more information about scalability, see the section “Ready for Mission-Critical Workloads.”

SQL Server 2008 R2 also will include an increase in price, which might affect your purchasing and licensing decisions. Microsoft is increasing the per-processor pricing for SQL Server 2008 R2 by 15 percent for the Enterprise Edition and 25 percent for the Standard Edition. There will be no change to the price of the server and CAL licensing model. The new retail pricing is shown in Table 1.

Licensing is always a key consideration when you are planning to deploy a new version of SQL Server. Two things have not changed when it comes to licensing SQL Server 2008 R2:

- The per server/client access model pricing for SQL Server 2008 R2 Standard and Enterprise has not changed. If you use that method for licensing SQL Server deployments, the change in pricing for the per-processor licensing will not affect you. That’s a key point to take away.
- SQL Server licensing is based on sockets, not cores. A socket is a physical processor. For example, if a server has four physical processors, each with eight cores, and you choose Enterprise, the cost would be \$114,996. If Microsoft charged per core, the

cost would be an astronomical \$919,968. Some of Microsoft’s competitors do charge per core, so keep that in mind as you decide on your database platform and consider what licensing will cost over its lifetime.

Buy Software Assurance (SA) now to lock in your existing pricing structure before SQL Server 2008 R2 is released, and you’ll avoid paying the increased prices for SQL Server 2008 R2 Standard and Enterprise licenses. If you’re planning to upgrade during the period for which you have purchased SA, you’ll save somewhere between 15 percent and 25 percent. If you purchase SA, you also will be able to continue to use unlimited virtualization when you decide to upgrade to SQL Server 2008 R2 Enterprise Edition.

One change that you should note: In the past, the Developer edition of SQL Server has been a developer-specific version that contained the same specifications and features as Enterprise. For SQL Server 2008 R2, Developer now matches Datacenter.

Upgrading to SQL Server 2008 R2

SQL Server 2000, SQL Server 2005, or SQL Server 2008 can all be upgraded to SQL Server 2008 R2. The upgrade process is similar to the one documented for SQL Server 2008 for standalone or clustered instances. A great existing resource you can use is the SQL Server 2008 Upgrade Technical Reference Guide, available for download at Microsoft.com.

You can augment this reference with the SQL Server 2008 R2-specific information in SQL Server 2008 R2 Books Online. Useful topics to read include “Version and Edition Upgrades” (documents what older versions and editions can be upgraded to which edition of SQL Server 2008 R2), “Considerations for Upgrading the Database Engine,” and “Considerations for Side-by-Side Instances of SQL Server 2008 R2 and SQL Server 2008.” As with SQL Server 2008, it’s highly recommended that you run the Upgrade Advisor before you upgrade to SQL Server 2008 R2. The tool will check the viability of your existing installation and report any known issues it discovers.

BI Enhancements in SQL Server 2008 R2

This guide focuses on the relational side of SQL Server 2008 R2, but Microsoft has also enhanced the features used for BI. Some of those improvements include:

- SQL Server 2008 R2 Parallel Data Warehouse, shipping later in 2010, makes data warehousing more cost effective. It can manage hundreds of terabytes of data and deliver stellar performance with parallel data movement and a scale out architecture.

Table 1: SQL Server 2008 R2 retail pricing

Edition	Per Processor (price in US dollars)	Per Server + Client Access Licenses (price in US dollars)
Standard	\$7,499	\$1,849 with 5 CALs
Enterprise	\$28,749	\$13,969 with 25 CALs
Datacenter	\$57,498	Not available
Parallel Data Warehouse	\$57,498	Not available

With the capability to leverage commodity hardware and the existing tools, Parallel Data Warehouse (formerly known as “Madison”) becomes the center of a BI deployment and allows many different sources to connect to it, much like a hub-and-spoke system. Figure 1 shows an example architecture.

- SQL Server 2008 R2 Enterprise and Datacenter ship with Master Data Services, which enables *master data management (MDM)* for an organization to create a single source of “the truth,” while securing and enabling easy access to the data. To briefly review MDM and its benefits, consider that you have lots of internal data sources, but ultimately you need one source that is the master copy of a given set of data. Data is also interrelated (for example, a Web site sells a product from a catalog, but that is ultimately translated into a transaction where an item is purchased, packed, and sent; that requires coordination). Such things as reporting might be tied in as well—essentially encompassing both the relational and analytic spaces. These concepts and the tools to enable them are known as MDM.
- PowerPivot for SharePoint 2010 is a combination of client and server components that lets end

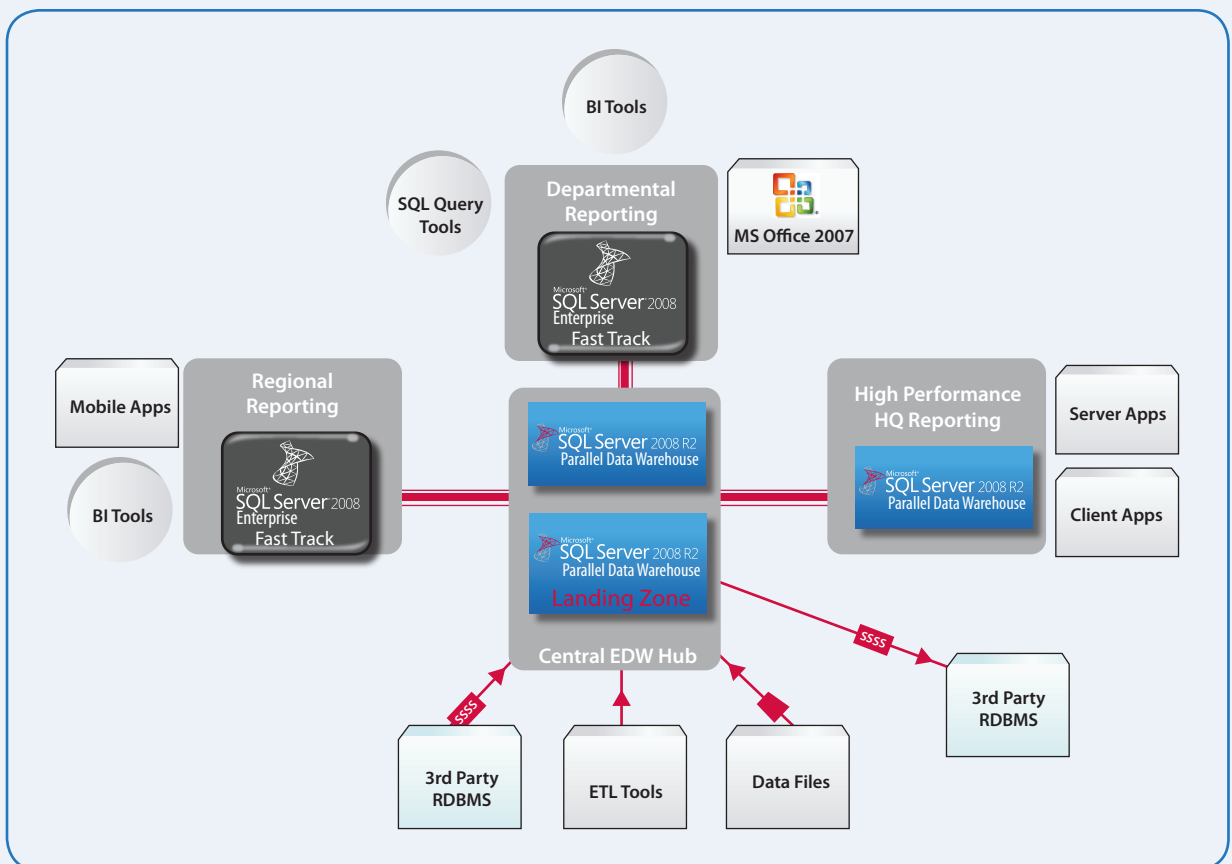
users use a familiar tool—Excel—to securely and quickly access large internal and external multidimensional data sets. PowerPivot brings the power of BI to users and lets them share the information through SharePoint, with automatic refresh built in. No longer does knowledge need to be siloed. For more information, go to <http://www.powerpivot.com/>.

- Reporting Services has been greatly enhanced with SQL Server 2008 R2, including improvements to reports via Report Builder 3.0 (e.g., ways to report using geospatial data and to calculate aggregates of aggregates, indicators, and more).

Consolidation and Virtualization

Consolidation of existing SQL Server instances and databases can take on various flavors and combinations, but the two main categories are physical and virtual. More and more organizations are choosing to virtualize, and choosing the right edition of SQL Server 2008 R2 can lower your costs significantly. A SQL Server 2008 Standard license comes with one virtual machine license; Enterprise allows up to four. By choosing Datacenter and licensing an entire server, you get the right to deploy as many virtual machines as you want on that server,

Figure 1: Example of a Parallel Data Warehouse implementation showing the hub and its spokes



whether or not the virtualization platform is from Microsoft. And as mentioned earlier, if you purchase SA for an existing SQL Server 2008 Enterprise license, you can enjoy unlimited virtualization on a server with SQL Server 2008 R2. However, this right does not extend to any major SQL Server Enterprise release beyond SQL Server 2008 R2. So consider your licensing needs for virtualization and plan accordingly for future deployments that are not upgrades covered by SA.

Live Migration is one of the most exciting new features of Windows Server 2008 R2. It's built upon the Windows failover clustering feature and provides the capability to move a Hyper-V virtual machine from one node of the cluster to another with zero downtime. SQL Server's performance will degrade briefly during the migration, but applications and end users will never lose their connection, nor will SQL Server stop processing transactions. That capability is a huge leap forward for virtualization on Windows. Live Migration is for planned failovers, such as when you do maintenance on the underlying cluster node; but it allows 100 percent uptime in the right scenarios. Although its performance will temporarily be affected in the switch of the virtual machine from one server to another, SQL Server continues to process transactions. Everyone may not need this functionality, but Live Migration is a compelling reason to consider Hyper-V over other platforms since it ships as part of Windows Server 2008 R2.

For deployments of SQL Server, SQL Server 2008 R2 supports the capability to SysPrep installations for an installation on a standalone server or virtual machine. This means that you can easily and consistently standardize and deploy SQL Server configurations. And ensuring that each SQL Server instance looks, acts, and feels the same to you as the DBA improves your management experience.

Note that if you're using a non-Microsoft platform for virtualizing SQL Server, it must be listed as part of the Server Virtualization Validation Program (<http://www.windowsservercatalog.com/svvp.aspx>). Check this list to ensure that the version of the vendor's hypervisor is supported. Also make sure to check Knowledge Base article 956893 (<http://support.microsoft.com/kb/956893>), which outlines how SQL Server is supported in a virtualized environment.

Management

SQL Server 2008 introduced two key management features: Policy-Based Management and Resource Governor. Policy-Based Management lets you define a set of rules (a policy) based on a specific set of conditions (the condition). A condition is made up of facets (e.g., Database); it then has a bunch of properties (such as AutoShrink) that can be evaluated. Consider this example: As DBA, you

do not want AutoShrink enabled on any database. AutoShrink can be queried for a database, and if it is enabled, it can either just be reported back as such and leave you to perform an action manually if desired, or if a certain condition is met (such as AutoShrink being enabled) and that is not your desired condition, it can be disabled automatically once it has been detected. The choice is up to you as the implementer of the policy. You can then roll out the policy and enforce it for all of the SQL Server instances in a given environment. With PowerShell, you can even use Policy-Based Management to manage SQL Server 2000 and SQL Server 2005. It's a great feature that you can use to enforce compliance and standards in a straightforward way.

Resource Governor is another feature that is important in a post-consolidated environment. It allows you as DBA to ensure that a workload will not bring an instance to its knees, by defining CPU and memory parameters to keep it in check—effectively stopping things such as runaway queries and unpredictable executions. It can also allow a workload to get priority over others. For example, let's say an accounting application shares an instance with 24 other applications. You observe that, at the end of the month, performance for the other applications suffers because of the nature of what the accounting application is doing. You can use Resource Governor to ensure that the accounting application gets priority, but that it doesn't completely starve the other 24 applications. Be aware that Resource Governor cannot throttle I/O in its current implementation and should only be configured if there is a need to use it. If it's configured where there is no problem, it could potentially cause one; so use Resource Governor only if needed.

SQL Server 2008 R2 also introduces the SQL Server Utility. The SQL Server Utility gives you as DBA the capability to have a central point to see what is going on in your instances, and then use that information for proper planning. For example, in the post-consolidated world, if an instance is either over- or under-utilized, you can handle it in the proper manner instead of having the problems that existed pre-consolidation. Before SQL Server 2008 R2, the only way to see this kind of data in a single view was to code a custom solution or buy a third-party utility. Trending and tracking databases and instances was potentially very labor and time intensive. To address this issue, the SQL Server Utility is based on setting up a central Utility Control Point that collects and displays data via dashboards. The Utility Control Point builds on Policy-Based Management (mentioned earlier) and the Management Data Warehouse feature that shipped with SQL Server 2008. Setting up a Utility Control Point and enrolling instances is a process that takes minutes—not hours,

days, or weeks, and large-scale deployments can utilize PowerShell to enroll many instances. SQL Server 2008 R2 Enterprise allows you to manage up to 25 instances as part of the SQL Server Utility, and Datacenter has no restrictions.

Besides the SQL Server Utility, SQL Server 2008 R2 introduces the concept of a data-tier application (DAC). A DAC represents a single unit of deployment,

or package, which contains all the database objects needed for an application that you can create from existing applications or in Visual Studio if you are a developer. You will then use that package to deploy the database portion of the application via a standard process instead of having a different deployment method for each application. One of the biggest benefits of the DAC is that you can modify it at any

Figure 2: Sample SQL Server Utility architecture

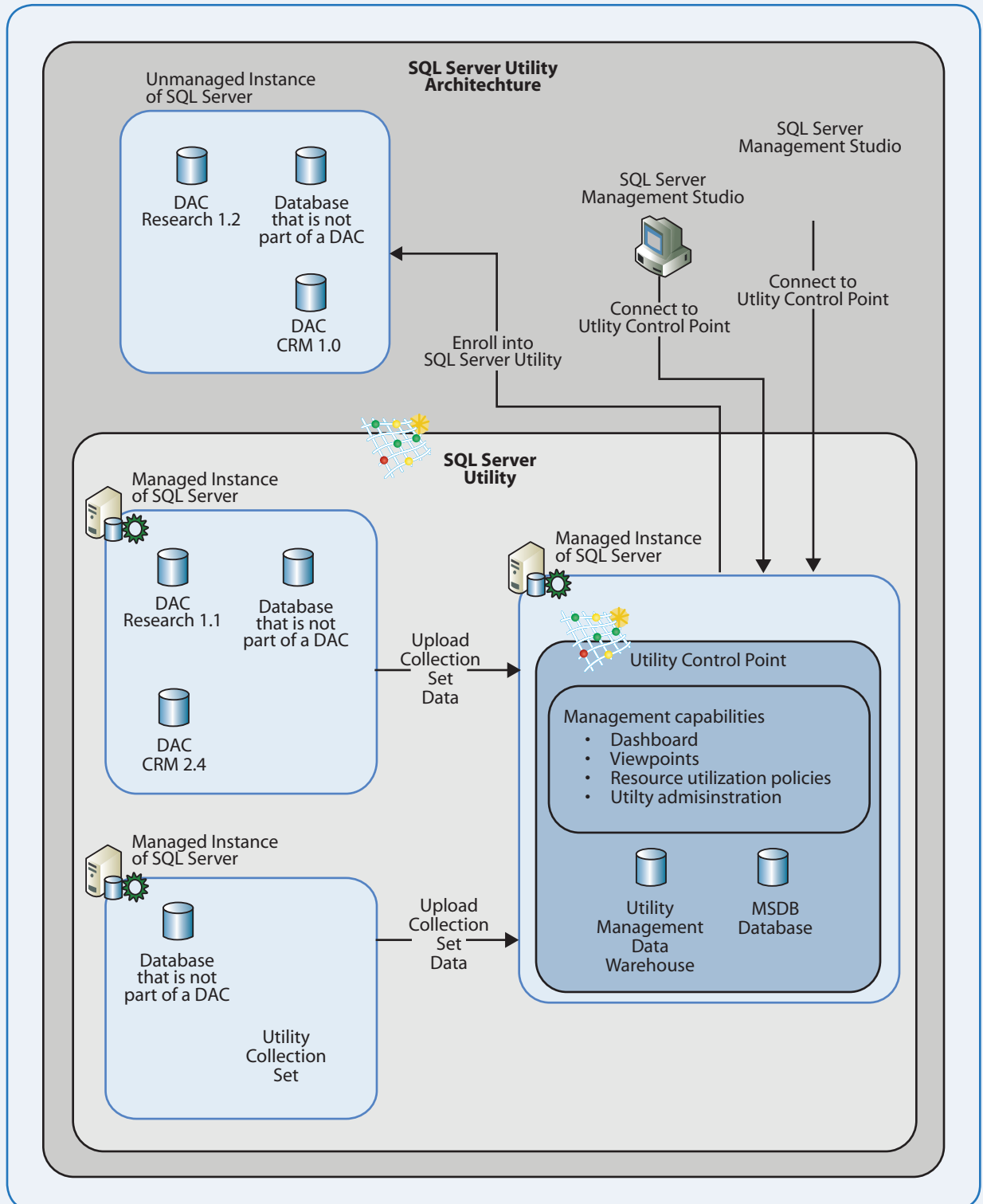


Table 2: The major differences between Standard, Enterprise, and Datacenter editions

	Standard	Enterprise	Datacenter
Max Memory	64GB	2TB	Maximum memory supported by Windows version
Max CPU (licensed per socket, not core)	4 sockets	8 sockets	Maximum memory supported by Windows version

time in Visual Studio 2010 (for example, adding a column to a table). In addition, you can redeploy the package using the same standard process as the initial deployment in SQL Server 2008 R2, with minimal to no intervention by IT or the DBAs. This capability makes upgrades virtually painless instead of a process fraught with worry. Certain tasks, such as backup and restore or moving data, are not done via the DAC; they are still done at the database. You can view and manage the DAC via the SQL Server Utility. Figure 2 shows what a sample SQL Server Utility architecture with a DAC looks like.

IMPORTANT: Note that DAC can also refer to the dedicated administrator connection, a feature introduced in SQL Server 2005.

Ready for Mission-Critical Workloads

Mission-critical workloads (including things such as consolidation) require a platform that has the right horsepower. The combination of Windows Server 2008 R2 and SQL Server 2008 R2 provides the best performance-to-cost ratio on commodity hardware. Table 2 highlights the major differences between Standard, Enterprise, and Datacenter.

With Windows Server 2008 R2 Datacenter, SQL Server 2008 R2 Datacenter can support up to 256 logical processors. If you're using the Hyper-V feature of Windows Server 2008 R2, up to 64 logical processors are supported. That's a lot of headroom to implement physical or virtualized SQL Server deployments. SQL Server 2008 R2 also has support for hot-add memory and processors. As long as your hardware supports those features, your SQL Server implementations can grow with your needs over time, instead of your having to overspend when you are initially sizing deployments.

SQL Server 2008 R2 Datacenter achieved a new world record of 2,013 tpsE (tps = transactions per second) on a 16-processor, 96-core server. This record used the TPC-E benchmark (which is closer to what people do in the real world than TPC-C). From a cost perspective, that translates to \$958.23 per tpsE with the hardware configuration used for the test, and it shows the value that SQL Server 2008 R2 brings to the table for the cost-to-performance

ratio. You can find the full results and information about TPC-E at <http://www.tpc.org>.

Note that Windows Server 2008 R2 is 64-bit only, so now is a good time to start transitioning and planning deployments to 64-bit Windows and SQL Server. If for some reason you still require a 32-bit (x86) implementation

of SQL Server 2008 R2, Microsoft is still shipping a 32-bit variant. If an implementation is going to use a 32-bit SQL Server 2008 R2 instance, it's recommended that you deploy it with the last 32-bit server operating system that Microsoft will ever ship: the original (RTM) version of Windows Server 2008.

Performance is not the only aspect that makes a platform mission critical; it must also be available. SQL Server 2008 R2 has the same availability features as its predecessor, and it can give your SQL Server deployments the required uptime. Add to that mix the Live Migration feature for virtual machines, mentioned earlier; and there are quite a few methods to make instances and databases available, depending on how SQL Server is deployed in your environment.

Don't Overlook SQL Server 2008 R2

SQL Server 2008 R2 is not a minor point release of SQL Server and should not be overlooked; it is a new version full of enhancements and features. If you are an IT administrator or DBA, or you are doing BI, SQL Server 2008 R2 should be your first choice when it comes to choosing a version of SQL Server to use for future deployments. By building on the strong foundation of SQL Server 2008, SQL Server 2008 R2 allows you as the DBA to derive even more insight into the health and status of your environments without relying on custom code or third-party utilities—and there is an edition that meets your performance and availability needs for mission-critical work, as well.

Allan Hirt has been using SQL Server in various guises since 1992. For the past 10 years, he has been consulting, training, developing content, speaking at events, and authoring books, whitepapers, and articles. His most recent major publications include the book *Pro SQL Server 2005 High Availability* (Apress, 2007) and various articles for *SQL Server Magazine*. Before striking out on his own in 2007, he most recently worked for both Microsoft and Avanade, and still continues to work closely with Microsoft on various projects. He can be reached via his website at <http://www.sqlha.com> or at allan@sqlha.com.