

Using SQL Server 2012 to **Decrease IT Cost**

By Michael K. Campbell



SQL Server 2012 represents a major release, one that provides a number of new features, options, tools, capabilities, and benefits not found in previous versions. SQL Server 2012 will also ship with an updated licensing model.

This eGuide provides an overview of key licensing changes and their implications and focuses on how advances and features provided in SQL Server 2012 can help provide businesses with the ability to decrease operating costs while simultaneously helping to enable increased performance, scalability, availability, and manageability.

Contents

SQL Server 2012 Licensing Changes	2
SKU Cleanup and SQL Server Business Intelligence Edition	2
Business Intelligence Edition and Changes to CALs.....	2
Licensing Moves to Cores—Instead of Sockets	3
Game Changing SQL Server 2012 Features.....	4
Partially Contained Databases	4
Indirect Checkpoints.....	5
Support for SMB File Shares.....	5
SQL Server AlwaysOn.....	5
SQL Server AlwaysOn Failover Cluster Instances	5
SQL Server AlwaysOn Availability Groups	5
Server Core.....	6
SQL Server 2012—Greater than the Sum of Its Parts	7
Call to Action	7

Michael K Campbell

Michael K Campbell is a contributing editor for *SQL Server Pro* and a consultant with years of SQL Server DBA and developer experience. He enjoys consulting, development, and creating free videos for www.sqlservideos.com.

(c) 2011 Microsoft Corporation. All rights reserved. This document is provided “as-is.” Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

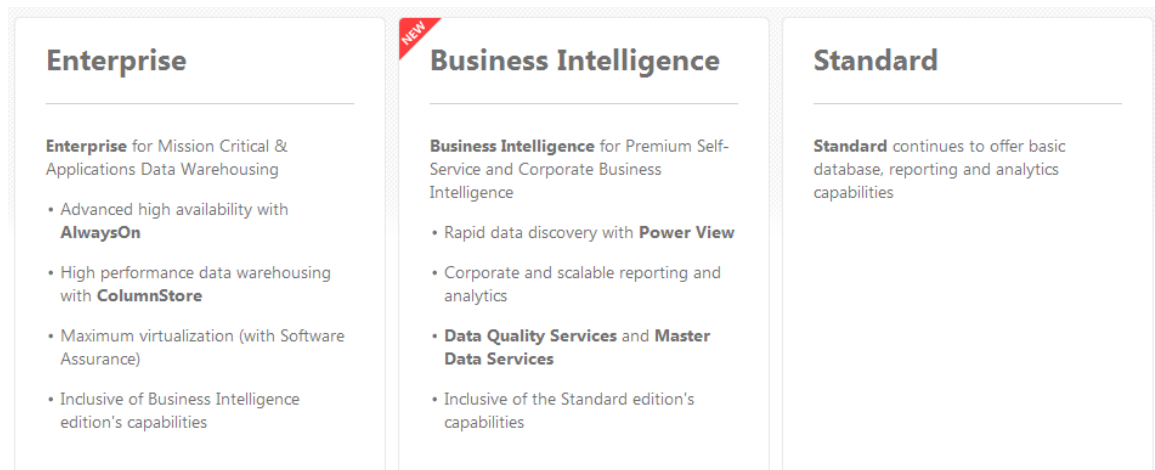
This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes. You may modify this document for your internal, reference purposes.

SQL Server 2012 Licensing Changes

Although few people enjoy reading about licensing, there are a few key changes to SQL Server 2012 that everyone should understand. (Of course, if you already know about these changes or don't care, then feel free to skip to the next section.) And even though it's impossible to summarize all the changes to SQL Server 2012 licensing within just a few paragraphs, key changes are as follows.

SKU Cleanup and SQL Server Business Intelligence Edition

With SQL Server 2012, Microsoft has standardized on three primary editions of SQL Server: Enterprise, Standard, and (a new entry) Business Intelligence Edition, which offers increased Business Intelligence capabilities beyond those in Standard Edition (Figure 1).



* The Web, Developer and Express editions will also be available in the SQL Server 2012 release.

Figure 1: SQL Server editions

Although SQL Server 2012 will also feature Express, Developer, and Web Editions (which will be licensed solely via Services Provider Licensing Agreements), SQL Server 2012 streamlines Data Center Edition into the Enterprise Edition and retires previous Workgroup and Small Business Editions. This move facilitates a more simplified offering.

Business Intelligence Edition and Changes to CALs

SQL Server 2012 Business Intelligence Edition can be licensed either in terms of processing power (i.e., the number of processors) or according to the number of assigned users (via an initial Server license plus Client Access Licenses—CALs). In this regard, SQL Server 2012 Business Intelligence Edition offers premium reporting and self-service BI features at a fraction of the cost of what would be required to

license entire servers using Enterprise Edition. With SQL Server 2012, CAL prices are also expected to increase by about 27 percent.

Licensing Moves to Cores—Instead of Sockets

With SQL Server 2012, server licensing switches from sockets to cores. Under this new model, licensing costs do not increase. Instead, the focus on cores instead of sockets enables a more precise measurement of processor usage for SQL Server workloads.

Accordingly, each processor-core license for SQL Server 2012 costs one-quarter of the socket-license cost for SQL Server 2008. SQL Server 2012 processor-core licenses are, in turn, sold in dual-core packs and processors must be licensed according to the total number of cores present on each socket. Additionally, each socket must also be licensed for a minimum of four cores.

In this way, quad-core processors for Enterprise and Standard Edition cost exactly as much to license with SQL Server 2012 as they do currently. However, as processors with higher core counts (such as hexa-core, or deca-core processors) provide a greater amount of processing power for SQL Server workloads, Microsoft's newer, and more precise, licensing model requires an increased number of licenses.

Obviously, this means that many higher-end servers will cost more to license. From Microsoft's standpoint, this is merely because these servers provide more compute power. Still, while it's true that no one enjoys paying more for anything, it would be a mistake to look at SQL Server 2012 solely in terms of licensing costs—especially because licensing costs are only one facet of overall ownership costs; not to mention that SQL Server 2012 ships with powerful new features and abilities that will help organizations decrease overall management costs while simultaneously helping them achieve greater uptime, availability, and business agility.

Game Changing SQL Server 2012 Features

With any new product release there are always incremental changes along with new features, tools, and benefits. SQL Server 2012 is no different in this regard, boasting a huge number of significant improvements and additions. However, while some new features and changes to the SQL Server database engine might initially appear beneficial in their own right, when taken together they offer truly synergistic, or game-changing, potential when it comes to helping enable more pro-active manageability, which translates into decreased IT costs.

Partially Contained Databases

One area of SQL Server manageability that DBAs have struggled with over the years centers on the fact that database users are defined at the database level, while logins into SQL Server are defined at the server level. This means that DBAs have to first copy logins to destination servers, then restore target databases as a means of hopefully avoiding some of the troubleshooting that typically follows as a process of re-linking “orphaned users” against their associated logins. Stated differently, copying and moving SQL Server databases can require appreciable administrative interaction, which doesn’t scale well.

With SQL Server 2012, a new feature known as Partially Contained Databases provides an application boundary that helps clearly distinguish between server-level resources and those defined, or contained, at the database level. Furthermore, with Partially Contained Databases, DBAs can define users at the database level instead of at the server level. In this way, Partially Contained Databases become much more portable than their traditional counterparts.

To enable Partially Contained Databases, DBAs must first explicitly enable database-level authentication at the server level (using `sp_configure`). Individual databases can then be set to use `CONTAINMENT = PARTIAL` (via the `CREATE` or `ALTER` database commands) to enable partial containment. Furthermore, SQL Server 2012 also introduces a new `CURRENT` keyword that you can use in place of a database’s name in `ALTER` statements—thus freeing databases from dependencies upon even their names—as a means of making Partially Contained Databases that much easier to manage. In similar fashion, collations defined at the database level also become portable, and SQL Server 2012 even introduces two new Dynamic Management Views (DMVs) that allow DBAs and developers to easily query Partially Contained Databases for dependencies and other potential problems that might endanger database portability.

Not only do Partially Contained Databases offer DBAs the ability to create highly portable databases that can be easily moved from server to server as needed to manage workload requirements, but DBAs also benefit from additional failover capabilities as well, because Partially Contained Databases become much less dependent upon underlying servers.

Indirect Checkpoints

To ensure optimal performance, SQL Server performs all database modification in memory (in conjunction with the transaction log) instead of writing each and every change to underlying data files as it occurs. Then, to prevent the recovery process from having to process too many operations from the log file should a crash occur, SQL Server periodically issues a CHECKPOINT command that forces in-memory modifications to disk along with corresponding adjustments to the transaction log.

In earlier SQL Server versions, timing for CHECKPOINT operations is largely controlled at the server level (via `sp_configure`) such that a single setting applies to all databases. With SQL Server 2012, DBAs can define a CHECKPOINT interval on a database-by-database basis via the `ALTER DATABASE` command. Not only are DBAs able to better control recovery time objectives at a more granular level, but these settings now become portable with the targeted databases as well.

Support for SMB File Shares

SQL Server 2012's ability to host databases on System Messaging Block (SMB) File Shares provides several exciting manageability benefits. First and foremost, while current SMB File Shares are not capable of delivering IO throughput and stability on par with SAN or direct-attach storage, the anticipated release of SMB 2.2 will provide vastly improved throughput and stability—making SMB storage capable of addressing mission-critical workloads. Until then, many so-called “secondary” databases that typically demand a disproportionate amount of DBA and Systems Engineer attention in terms of provisioning and management can be safely deployed to SMB File Shares as a means of decreasing management costs. Furthermore, by making secondary databases Partially Contained, DBAs can easily shunt databases from one server to another (for workload balancing purposes) by merely detaching and reattaching files via their UNC file path.

SQL Server AlwaysOn

With SQL Server 2012, the ability to achieve increased availability, redundancy, and scale-out capabilities becomes significantly easier than before. To this end, while SQL Server 2012 still fully supports the deployment of SQL Server to Windows Server Failover Clusters (just as before), it also offers two exciting new options that require significantly less configuration and complexity, while simultaneously enabling increased features and capabilities over traditional clustering options.

SQL Server AlwaysOn Failover Cluster Instances

Unlike full SQL Server Cluster deployments, Failover Cluster Instances (FCIs) represent a decreased manageability option that allows DBAs and systems administrators to deploy a single SQL Server instance (instead of an entire Server and any or all instances therein) as a high-availability solution. In addition to offering significantly less complexity than traditional clustering options, FCIs can also span different subnets, facilitating improved disaster recovery options through increased redundancy while also enabling improved support for remote availability.

SQL Server AlwaysOn Availability Groups

Also built atop Windows Server Failover Cluster technology, AlwaysOn Availability Groups do not require the added complexity of a shared (or virtual) machine name, IP address, or shared storage.

Instead, they're much more akin to a more powerful and intelligent version of the SQL Server Database Mirroring functionality that they replace. They offer the ability to define associated groups of databases that will failover as a group instead of singly, thus helping DBAs preserve application dependencies. Likewise, in addition to providing much simpler management and failover capabilities, Availability Groups also provide improved scalability thanks to their ability to support multiple mirroring destinations (or facets), including read-only copies that can be used for scale-out, reporting, and improved disaster recovery capabilities.

Server Core

Server Core's reduced management overhead is well understood by IT professionals. With SQL Server 2012, organizations can finally deploy SQL Server to Server Core hosts with only a few, obvious, limitations, such as the lack of support for GUI tools. Moreover, as Server Core deployments benefit from smaller overall footprints that require less frequent patching, the use of SQL Server 2012 on Server Core instances will help boost overall uptime and availability. In this regard, Server Core deployments represent a very powerful, game-changing, option in the sense that they can be configured and deployed as SQL Server workload hosts in much the same way that virtualization hosts can be configured to handle typical, virtualized, Windows workloads.

More importantly, when the manageability benefits of SQL Server 2012 deployed on Server Core are combined with other manageability benefits such as those afforded by Partially Contained databases, support for File Shares, or merely an easier ability to enable failover and scalability through AlwaysOn capabilities, it's easy to see how SQL Server 2012 provides truly synergistic improvements.

SQL Server 2012—Greater than the Sum of Its Parts

On their own, the features and benefits described above enable compelling manageability benefits that can help decrease overall ownership and operational costs. However, when combined, these features make it possible to see how SQL Server 2012 not only embraces the current IT trend of consolidation and virtualization (as a means of helping increase manageability and decrease costs), but how it takes these same benefits to the next level by enabling DBAs and systems administrators to treat many SQL Server databases as workloads that can be more readily deployed, consolidated, and managed upon underlying SQL Server hosting platforms—or SQL Servers.

Additionally, while the features described above help facilitate decreased IT costs, they don't address other big wins provided by SQL Server 2012 in terms of improved developer productivity, increased data protection (in the form of helping to improve security and policy functionality and through the introduction and extension of SQL Server Master Data Services and SQL Server Data Quality Services). Nor do they touch upon the new features in SQL Server Reporting Services, SQL Server Integration Services, or SQL Server Analysis Services which all help contribute to increased insight, better overall business agility, and decreased management costs.

Call to Action

To learn more about SQL Server 2012 Licensing or Edition changes, visit www.microsoft.com/sql/. For a technical overview of what is new in SQL Server 2012 (as of Release Candidate 0), visit [http://msdn.microsoft.com/en-s/library/bb500435\(SQL.110\).aspx](http://msdn.microsoft.com/en-s/library/bb500435(SQL.110).aspx)

