

## CHAPTER ②

# Planning your SQL Server consolidation

*You've reviewed the benefits and decided to consolidate your SQL Server databases. But how should you approach the project?*

Sponsored by

### INSIDE

- 02 Plan early
- 03 The six steps to consolidation
- 05 Identify candidate servers and databases
- 07 Review version, software and hardware dependencies
- 14 Consolidate on 32-bit vs. 64-bit
- 15 Consolidate on single vs. multiple instances

# Planning your SQL Server consolidation

*By Hilary Cotter*

In [Chapter 1](#), I explained when and how to make the case for SQL Server consolidation — the centralization of user databases from a multitude of SQL Servers in dispersed environments. I reviewed the many benefits of consolidation, including improved management, more efficient use of hardware resources and licenses, and improved monitoring.

Now that you have decided to consolidate your SQL Server databases, I'll look at how to approach the project.

Despite the benefits outlined, SQL Server consolidation is not always an easy sell with business units. Business owners are reluctant to buy into consolidation

projects because doing so may mean losing control of databases moved from staff to central management, losing database visibility once moved to the data center, degraded SQL Server performance, or lessened support or functional enhancements from the new consolidation team.

Your consolidation team must keep these concerns in mind and plan carefully to deliver a successful consolidation project that provides higher availability, better performance and more proactive management. With multiple user databases being consolidated or stacked onto a single SQL Server, performance problems on even a single user database may now affect all user

**“While it is possible from a technical standpoint to consolidate an entire enterprise in one, mass migration, a better approach is to move groups of user databases into a consolidated environment in phases.”**



Time spent planning your consolidation is critical to success.

databases. In the past this type of problem could be insulated on a single SQL Server.

While it is possible from a technical standpoint to consolidate an entire enterprise in one, mass migration, a better approach is to move groups of user databases into a consolidated environment in phases. Then you can monitor and stabilize the newly consolidated SQL Servers before migrating the next phase. By rolling out your consolidation in waves, your project team will be well practiced and develop a deep understanding of problems that arise. Attempting an all-at-once, enterprise-wide consolidation can quickly overwhelm the project team, take longer to stabilize and damage user confidence in the consolidation effort.

Time spent in planning your consolidation is critical to a successful deployment.

## Map out your consolidation

*Planning a consolidation is a mammoth task.*

*Simplify the project by breaking it into individual components.*

① **Create a consolidation methodology.**

② **Analyze candidate user databases, servers, storage, hardware and applications.**

③ **Test in a test environment.**

④ **Deploy in a production environment.**

⑤ **Monitor and stabilize.**

⑥ **Repeat for the next group.**

→ *In Chapter 3 I will delve into the specifics for each component. In this chapter, I'll provide an overview of each one in turn.*

### STEP 1: CREATE A CONSOLIDATION METHODOLOGY

To perform a successful enterprise-wide SQL Server consolidation you must first define goals for your consolidation team and the client, the business owner of user databases. These goals will vary depending on your consolidation approach: Consolidating on a virtual machine, stacking SQL Server instances, using a storage area network ([SAN](#)), etc.

It is critical that the consolidation team negotiate in advance realistic service level agreements ([SLAs](#)) with the client. These SLAs will not only set expectations for availability, support, change control and

monitoring, but also performance. An SLA that sets supportable expectations goes a long way in building confidence in the consolidation effort.

Mission-critical applications should be identified. Their SLAs tend to be more stringent than other SLAs, either requiring you not to consolidate these applications or carefully plan to ensure SLAs can be met or exceeded in the consolidated environment. Standards should be applied and those applications should be brought under the consolidation teams' own-

ership and control.

Another reason to negotiate SLAs in advance is to avoid scope creep, in which your consolidation team would have to solve unexpected performance problems and enhance functionality.

Your team must consider various scenarios when working with SLAs. For instance, someone may discover poorly performing applications while identifying databases that are good candidates for consolidation. Ideally clients should be asked to take these applications back for optimization. If your team elects to optimize, you'll be taking ownership of any performance problems or bugs that arise in the future. The wise option is to merely identify and return these databases to the business owners and specify this action in the SLA.

If the business units are unwilling or unable to reclaim and

**“An SLA (service level agreement) that sets supportable expectations goes a long way in building confidence in the consolidation effort.”**

# Database Utility™ for SQL Server

**SAVE BIG with the best consolidation solution for SQL Server:**

- ▶ **REDUCE SERVER COUNTS BY AT LEAST 50%**
- ▶ **REDUCE MANAGEMENT TIME BY UP TO 75%**
- ▶ **SAVE UP TO 70% ON YOUR TOTAL COST OF OWNERSHIP OVER A THREE-YEAR PERIOD**

**See for yourself!**

Click on the button below to estimate your own ROI. Enter data about your existing SQL Server environment and add up the savings you can realize by consolidating with PolyServe.

**CALCULATE MY SAVINGS**

 **POLYSERVE™**

[www.polyserve.com](http://www.polyserve.com)

optimize these SQL Servers, migrate them into your data center and enforce standards as much as possible, but do not consolidate these databases with another SQL Server. Consolidating a poor-performing user database may degrade the performance of all other user databases on that SQL Server.

Once the SLA is negotiated, your consolidation team should create a schedule that breaks down the enterprise-wide plan into phases.

The first phase should involve departments with the least complex user databases. That gives team members a chance to practice before tackling the more difficult consolidations. This phased approach also teaches them to be more adept at juggling user databases between consolidated SQL Servers as database loads change over time. For example, as a particular user database grows,

it may degrade the performance of all user databases on that consolidated SQL Server. On the other hand, as the life cycle for a particular application winds down, the resource requirements of that user database may decline and make it viable for a move to a lower horsepower server.

Test scripts should be created to help survey existing SQL Server applications. That will familiarize team members with performance monitoring and SQL Server Profiler to capture and replay representative loads and monitor the consolidated solution.

The consolidation team should also break into specialized groups to simplify monitoring the consolidated solution. Once members of the consolidation team understand their individual tasks and prepare for the consolidation, the next step is analysis

## STEP 2: ANALYZE

In the analysis stage, the consolidation team will take stock of each SQL Server and its user databases to determine their performance characteristics, resource requirements, dependencies and how to migrate them. In the consolidated environment multiple-user databases will be grouped or stacked together based on performance characteristics, internal and external dependencies, SLAs and versions. They will be grouped onto individual SQL Servers, also known as SQL Server stacks. There are three parts to analyzing candidate SQL Servers and their user databases for consolidation.

① Analyze data access and usage patterns, and then analyze the SQL Server environment where the user database is installed.

② Duplicate these patterns in the consolidated environment so the user experience remains the same, if not better.

③ Investigate any other consolidation opportunities beyond the user databases. Can you consolidate the SQL Servers into a cluster? Can you consolidate storage (disk arrays and storage area networks)? Does it make sense to consolidate

applications such as SharePoint Team Services and Windows SharePoint Services, which tend to proliferate?

LET'S LOOK AT the specific aspects of the analysis stage.

You must have a good understanding of the usage and access patterns for the user databases being considered for consolidation. For example, some usage and access patterns are cyclical. Some databases are accessed 24x7 while others are used only 9-to-5 Monday through Friday. Databases experience peak times on an hourly, daily, weekly, monthly, quarterly or yearly basis.

SharePoint databases typically experience uniform usage during business hours, whereas payroll databases experience greater loads during weekly, biweekly or bimonthly payroll cycles. Accounting databases may experience peak loads

## How to analyze candidate SQL Servers and databases

1. Analyze data access and usage patterns
2. Duplicate patterns in the consolidated environment
3. Investigate other consolidation opportunities

SQL Server 2005 is the ideal platform for SQL Server consolidation because it offers improved functionality, performance, memory accessibility and clustering abilities.

quarterly or yearly. It is important to understand the cyclical nature of each load so that you can capture a representative load. Then, you can use that load in your test environment and plan accordingly for peak load times without crippling other user databases on your consolidated SQL Server stack.

Client interviews go a long way in determining the cyclical nature of the data access and data usage patterns. They will be able to identify not only peak usage times, but also the time of peak load (i.e., when batch jobs put extreme loads on the SQL Server) and help estimate anticipated growth.

Run basic health checks, such as DBCC CheckAlloc, and review the SQL Server error logs, the log directory for debug dumps and the application log for anomalies, any of which might point to problems or problem-usage patterns. With this knowledge your consolidation team can capture a representative load using SQL Profiler and replay it as a test in the consolidated solution. Then they can replay loads from several individual SQL Servers simultaneously on the test solution. Capture performance metrics on the original system and

compare those with the metrics obtained on the test consolidated SQL Server to quantify the performance improvement. Chapter 3 will detail the specific performance counters that should be gathered.

Many dependencies can make a user database a poor candidate for consolidation. It is critical that these dependencies are spotted in advance so you can plan for them or even work around them.

There are several classes of dependencies:

- Version
- Software
- External/hardware
- Internal

### VERSION DEPENDENCIES

SQL Server 2005 is the ideal platform for SQL Server consolidation because it offers improved functionality, performance and memory accessibility, and the ability to

## Dependencies that make or break database candidates for consolidation

→ Version

→ Software

→ External/hardware

cluster with SQL Server 2005 Standard Edition. Run SQL Server 2005 Upgrade Advisor to identify problems you may have upgrading your user database from previous versions. Or, consult its help file under the section marked Resolving Upgrade Issues. User databases that can't be upgraded to SQL Server 2005 or must run on previous versions of SQL Server should be consolidated on their own stacks: SQL Server 7 databases should be consolidated

on SQL Server 7 instances, SQL Server 2000 on SQL Server 2000 instances, etc.

You may be forced to consolidate user databases on previous versions of SQL Server because the vendor may not have certified its product to run on SQL Server 2005.

It is possible to set database compatibility mode to a previous version of SQL Server to

solve differences in certain database behaviors among versions. This will not solve any problems mentioned in the Upgrade Advisor, but it may solve problems with user applications that perform version checks.

Also watch for dependencies on SQL Server 2000 32-bit. For example, SQL Server 2000 64-bit does not support Data Transformation Services (DTS), Reporting Services, Notification Services, English Query, SQL Mail, some SQL Full-Text Search iFilters and Jet Support. This again will force consolidation on a SQL Server 2000 32-bit stack. There are no incompatibilities between SQL Server 2005 32 and 64-bit.

### SOFTWARE DEPENDENCIES

Software dependencies include applications that run on SQL Server and must be installed on the consolidated SQL Server environment. Examples include

File Transfer Protocol ([FTP](#)) and Internet Information Servers ([IIS](#)) (for Reporting Services, SQL Server CE and SQL XML ISAPI), SQL Server CE, SQL [Full-Text](#) search iFilters, [ODBC](#) drivers, OLE-DB providers and Microsoft Data Access Components (MDAC). While surveying user databases you must survey the Windows server as well to check for installed applications and drivers that SQL Server, linked servers, jobs, batch files or stored procedures might use.

The Add/Remove applet does provide information about installed programs and makes a good starting point for your inventory of third-party software installed on the source SQL Server environment. Also check the scheduler (Scheduled Tasks in Control Panel) and SQL Server jobs to watch for references to third-party software.

To get a list of installed OLE-

“If you consolidate in different geographical regions you will need high-speed links between the client system and the consolidated SQL Server to minimize the impact of the network hop.”



DB providers issue a call to `sp_enum_oledb_providers` on the candidate SQL Server and compare this with the list returned on the server where you are consolidating.

Also check for merge replication custom resolvers. You can obtain a list by issuing a call to `sp_enumcustomresolvers`.

### EXTERNAL AND HARDWARE DEPENDENCIES

Hardware dependencies involve hardware that is essential to the operation of SQL Servers or user databases. This includes 32 or 64-bit hardware, memory, solid-state memory drives, SANs and even hard-drive lettering. Hardware dependencies usually won't allow you to downgrade. For example, it is unlikely that you would be able to consolidate user databases running on a 64-bit operating system to a 32-bit operating system due to performance

limitations. Stored procedures, batch files and replication processes may also expect to use a specific drive letter (i.e. G drive), path or share, which may not be on your consolidated SQL Server. Also watch out for use of the Q drive, a drive letter reserved for clustered servers.

Keeping an eye out for these hardware dependencies will allow you to plan ahead, and rewrite jobs and stored procedures to reference the new drive lettering scheme. Also examine the network infrastructure. If you consolidate in different geographical regions you will need high-speed links between the client system and the consolidated SQL Server to minimize the impact of the network hop.

Other external dependencies include batch and command files, scripts ([SQL](#), [VBScript](#), [JavaScript](#), and [Perl](#)), directo-

*“A successful consolidation project requires that you scrupulously gather a wide variety of information about the candidate SQL Servers and their databases. This information will help you determine how to consolidate user databases and provide before and after metrics.”*

ries, Component Object Model (COM) objects and NT accounts. Again check the jobs in Job Scheduler, Task Scheduler and stored procedures that reference any external dependencies.

# Database Utility™ for SQL Server

## A BETTER APPROACH FOR SQL SERVER CONSOLIDATION

Now you can avoid the pitfalls of virtual machine strategies that limit performance and increase system failure risk. Through PolyServe's unique approach to server virtualization for SQL Server consolidation, you can finally reduce the cost and time required to manage many mission-critical SQL Server databases.

## THE ONLY SQL SERVER CONSOLIDATION SOLUTION THAT...

- ▶ PROVIDES FULL BARE-METAL SYSTEM PERFORMANCE
- ▶ ASSURES FULL DATABASE AND SERVER FAIL-OVER WITHOUT ADDED COST OR COMPLEXITY
- ▶ ACTUALLY SIMPLIFIES MANAGEMENT OF SQL SERVER ARCHITECTURE

## Database Utility™

Finally, you have a dedicated solution that enables multiple SQL Server instances to be consolidated onto fewer servers. Powerful benefits of the Database Utility include:

- **Improved Uptime:** High Availability services that are guaranteed for all SQL Server instances and can be easily configured to run on a mix of servers.
- **Improved Performance:** SQL Server still runs directly on the Windows Server OS; no virtual machine overhead, no CPU limitations, just bare-metal performance.
- **Improved Management:** Matrix Manager provides a single cluster management point for monitoring cluster-wide IO performance, storage capacity and usage and server health. Establish multi-tiered failover targets for guaranteed availability.
- **Improved DBA efficiency:** Push installs, hot-fixes and updates of one instance to all failover targets automatically. Re-host online instances to another server via “drag and drop” to perform server maintenance or capacity upgrades. Significantly simplify and streamline backups.

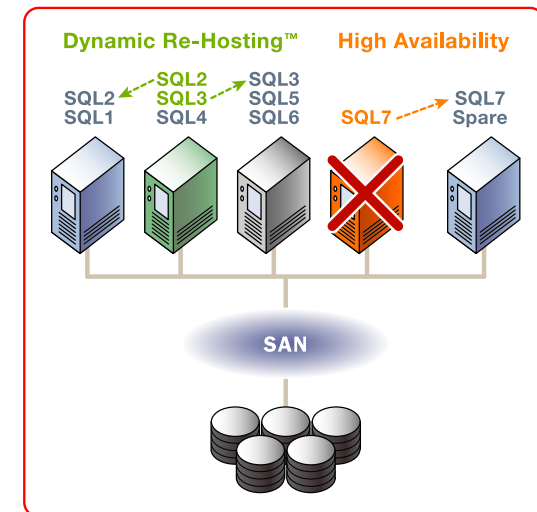
## Learn more:

[Visit the Database Utility resource page](#)

[View an on-demand Database Utility webcast](#)

[Download the Database Utility white paper](#)

## DATABASE UTILITY™ FOR SQL SERVER



[www.polyserve.com](http://www.polyserve.com)

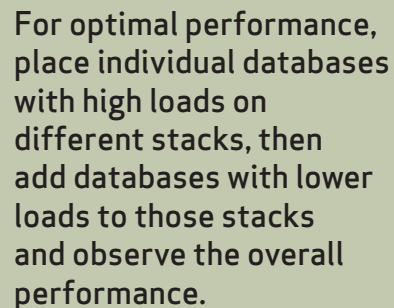
**INTERNAL DEPENDENCIES**

Your consolidation team must also pay close attention to internal dependencies of user databases on their current SQL Servers. These internal dependencies are:

- Server-wide settings
- MSSearch (SQL Full-Text Search)
- Named instances
- Server collations and database collations
- Accounts
- Name collisions
- User stored procedures in the master database
- Extended stored procedures
- CLR assemblies
- Replication
- Model db
- Jobs
- DTS packages
- Database dependencies
- Startup objects

Because internal dependencies have the lowest visibility and present the greatest stumbling blocks to consolidation, we will be looking in great depth at them in Chapter 3.

As you can tell, a successful consolidation project requires that you scrupulously gather a wide variety of information about the candidate SQL Servers and their databases. This information will help you determine how to consolidate user databases and provide before and after metrics.



**For optimal performance, place individual databases with high loads on different stacks, then add databases with lower loads to those stacks and observe the overall performance.**

**STEP 3: TEST**

Once you complete your analysis on the user databases, you must carefully determine which user database will be placed on which stack.

The following situations will force you to consolidate on different instances:

- User databases with unresolvable name conflicts
- User databases that make heavy use of tempdb
- User databases that come from SQL Servers that have non-standard collations

I provide a more complete list later in this chapter in the section entitled “Consolidating on single instances versus multiple instances.”

In some cases you will be forced to consolidate on a SQL Server running the previous version(s), often called a version stack. These include situations where you have version

dependencies or excessive use of SQL Full-Text Search.

Once you have determined which databases can be placed on which stack, carefully examine the data access and data usage patterns. This will help you group user databases on consolidated SQL Servers to provide optimal performance for that group. For example, place individual user databases with high loads on different stacks, then add user databases with lower loads to these stacks, and observe the overall performance. By juggling in this manner you will receive optimal throughput for your hardware dollar.

Carefully document any changes that you have made to the default SQL Server and database settings.

#### STEP 4: DEPLOY

After you have determined optimal placement of user databases on the consolidated SQL Server stacks, migrate user databases one by one from their current production environment to the new consolidated topology. After you migrate each user database, monitor the performance impact to ensure that the consolidated system remains stable with the introduction of these new user databases.

There is a multitude of ways to migrate user databases to the consolidated environment. One popular method is to detach each database from its current environment, copy it to the consolidated environment and attach it there.

However, replication will prevent you from detaching published databases. If your database is published, it is best to drop replication, move the

databases and then rebuild the publications and subscriptions. On SQL Server 2000 databases with full-text catalogs you will have to manually copy the database and the full-text catalogs to the consolidated environment as outlined by [Microsoft support](#). It is possible to use the Copy Database wizard, but it is simply not scalable for anything beyond small databases.

Before opening consolidated databases to the business owners, run warm-up scripts to ensure the first user experience in the new consolidated environment will not be a sluggish one.

**The goal of monitoring is not only to be proactive with problems, but to deliver a stabilized, consolidated solution.**

**STEP 5: MONITOR AND STABILIZE**

Even with the most careful analysis, testing and benchmarking, you will find surprises in the consolidated environment. In the earlier stages of the consolidation process the project team will become adept at providing optimal performance by monitoring and juggling user databases among consolidated SQL Servers.

Once a consolidated SQL Server goes into production, these monitoring and juggling skills will become invaluable. While it will be difficult to juggle some of the larger user databases, especially if they have external dependencies (such as replication or full-text search), some of the smaller user database can be moved easily should conditions warrant and you have a maintenance window.

Monitoring should allow you

to be highly proactive in managing each consolidated SQL Server, particularly looking for outages and performance problems. Many of the commercial monitoring programs are ideal for this, such as BMC Software, Inc. Patrol, Microsoft MOM and Quest Software, Inc. Spotlight.

The goal of monitoring is not only to be proactive with problems, but to deliver a stabilized, consolidated SQL Server solution that offers high performance and availability to all user databases and, more importantly, a solution that meets or exceeds negotiated SLAs. A stabilized environment is defined as an environment that can adequately service all loads with little to no performance degradation; from a monitoring perspective it is a quiet server that does not raise any performance alerts.

**STEP 6: REPEAT**

Once the first phase of consolidated SQL Servers is stabilized, migrate second-phase databases to the next consolidation group. Lessons learned in the first phase will accelerate subsequent phases.

**A stabilized environment is defined as an environment that can adequately service all loads with little to no performance degradation; from a monitoring perspective it is a quiet server that does not raise any performance alerts**

## Consolidating on 32-bit vs. 64-bit platforms

There are many advantages to consolidating on a 64-bit architecture — so much so that 32-bit is swiftly becoming obsolete. Mainly the operating system and applications are able to access more memory and faster performance: 32-bit applications can only access 4 GB of virtual memory, whereas 64-bit applications running on SQL Server 2005 64-bit Windows Server 2003 Enterprise Edition can access up to 1 TB.

Which SQL Server 2000 applications you can access with a 64-bit architecture is limited. By default, SQL Server 2000 Enterprise Edition is able to access 2 GB of virtual memory out of the box. With the 3-GB switch, SQL Server 2000 64-bit can access 3 GB of virtual memory. Using Addressing Windows Extensions (AWE), SQL Server 2000 64-bit Enterprise Edition on Windows Server 2003 can place its data cache in memory above the 4-GB limit in non-addressable memory. All other caches

(system data structures, user, stored procedure, buffer, log, etc.) would be squeezed into 1 GB.

The architecture of the 64-bit processor also allows for more efficient processing thanks to features including large on-die caches, larger throughput system bus, higher core frequencies and a larger number of registers.

Certain classes of workloads can take advantage of 64-bit processors, including those that are memory intensive and use large procedure caches, large

### Reasons to consolidate on multiple SQL Server instances


1. Improved throughput
2. Name collision prevention
3. Multi-instance clustering
4. More efficient use of procedure cache
5. Segregated development environments
6. Logistics
7. Segregated user databases
8. Collation problems addressed

cursors, large sorts and large numbers of connections.

Typical 64-bit applications include:

- Server consolidation with large numbers of databases on a single server and large numbers of procedures
- High-volume [OLTP](#)
- Large [OLAP](#)
- Simultaneous queries with large table joins

If you can't consolidate on SQL Server 2005 Enterprise Edition 64-bit, you should consolidate on SQL Server 2000 Enterprise Edition 64-bit. Keep in mind the limitations on SQL Server 2000 discussed above in the version dependencies section.



By default, SQL Server 2000 Enterprise Edition is able to access 2 GB of virtual memory.

## Consolidating on single vs. multiple instances

There are advantages to consolidating on multiple SQL Server instances. I'll touch on some of those reasons.

**IMPROVED THROUGHPUT** This MSDN document illustrates the [inflection point for transactions](#) per database peaks on a single instance at about 12,000 transactions per minute with 700 databases. The inflection point actually occurs at 55,000 transactions per minute on a multi-instance server with 500 databases per instance.

**NAME COLLISION PREVENTION** When you consolidate user databases from different SQL Servers you may find name collisions with user databases, account names, alerts, etc. Often the best way to prevent name collisions is to place the user databases with the name conflicts on different instances.

**MULTI-INSTANCE CLUSTERING FOR MULTIPLE INSTANCES** SQL Server 2000 and 2005 support multiple copies of SQL Server running concurrently on the same machine -- multiple instances. Multi-instance clustering uses multiple instances: Each instance runs on a different node in the multiple instance cluster and each node is "active." Should a node in the multiple-instance cluster fail, the instance running on that node will fail over to another active node and the resources it was using on the now failed node move to the other node. It's available in SQL Server 2000, so why not take advantage of it and increase your availability?

**MORE EFFICIENT USE OF PROCEDURE CACHE** Large numbers of stored procedures can cause memory pressures on a single instance. Multiple instances reduce these memory pressures.

**SEGREGATED DEVELOPMENT ENVIRONMENTS** Faults on one instance won't affect other development instances.

**LOGISTICS** You can dedicate instances to business units and geographic areas.

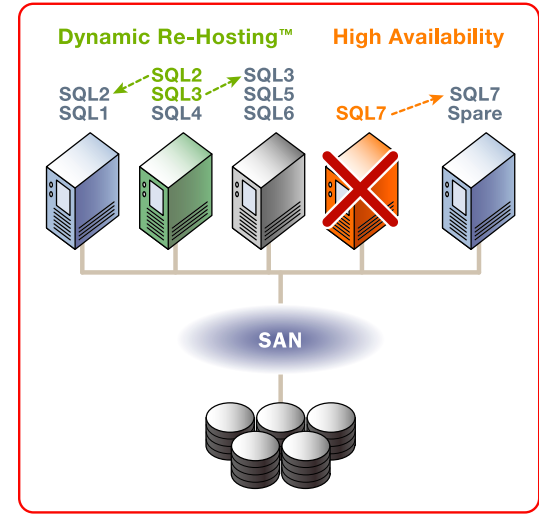
**SEGREGATED USER DATABASES** Multiple-user databases that make heavy use of tempdb will degrade performance or the entire instance. By putting these user databases on different instances you get improved overall performance.

**COLLATION PROBLEMS ADDRESSED** User databases that come from SQL Server 2000

servers with non-standard collations are problematic. Please refer to the section entitled [Mixed Collation Environments](#) in Books Online for more information on possible complications.

Keep in mind that not all applications are instance-aware; for example some applications may connect to SQL Server using dblib, which was written for previous versions of SQL Server and did not support instances. There is a workaround using the client networking utility on the client that runs the application to make the named instance look like a default instance. If the server name is hard coded into the application, you will have to connect to a consolidated SQL Server of the same name and place the user database on the default instance. Also keep in mind that each instance requires a separate license or set of licenses.

**DATABASE UTILITY™ FOR SQL SERVER**



**A better approach for SQL Server consolidation**

Avoid the pitfalls of virtual machine strategies that limit performance and increase system failure risk. Through PolyServe's unique approach to server virtualization for SQL Server consolidation, you can finally reduce the cost and time required to manage many mission-critical SQL Server databases.

**Learn more:**

[Visit the Database Utility resource page](#)

[View an on-demand Database Utility webcast](#)

[Download the Database Utility white paper](#)



[www.polyserve.com](http://www.polyserve.com)



## Summary

*As you can see, SQL Server consolidation requires careful planning, analysis and monitoring. Your consolidation team must be highly disciplined to closely follow the plan and instructions, take a rigorous approach to monitoring and analysis, and be careful not to allow scope creep. They must be vigilant in fighting the tendency to fix code-generated performance problems.*

I took a long look at all of the dependencies that you will have to take into account when embarking on a consolidation project. The team must watch for these potential issues and work around them as needed.

I also briefly touched on testing deployment and how to monitor and stabilize a solution in the test and production environments. Chapter 3 of this e-book will cover this in considerable depth, detailing the steps entailed in each process.

## ABOUT THE AUTHOR

Hilary Cotter has been involved in IT for more than 20 years as a Web and database consultant. Microsoft first awarded Cotter the Microsoft SQL Server MVP award in 2001. Cotter received his bachelor of applied science degree in mechanical engineering from the University of Toronto and studied economics at the University of Calgary and computer science at UC Berkeley. He is the author of a book on SQL Server transactional replication and is currently working on books on merge replication and Microsoft search technologies.