

Windows[®] IT Pro

How to Guide: SQL Server[®] 2005 Clustering

By Randy Dyess

Edited with permission from *SQL Server Magazine*.
Copyright © 2008 Penton Media, Inc.
All rights reserved.



Third-party information brought to you courtesy of Dell.

THIS WHITE PAPER IS FOR INFORMATIONAL PURPOSES ONLY, AND MAY CONTAIN TYPOGRAPHICAL ERRORS AND TECHNICAL INACCURACIES. THE CONTENT IS PROVIDED AS IS, WITHOUT EXPRESS OR IMPLIED WARRANTIES OF ANY KIND.

→ Contents

Introduction.....	1
Background on SQL Server Clustering: What Is It?	1
What types of clusters are there?	2
Single-Node Clusters	2
Multi-Node Clusters.....	3
What Are the Top 3 Challenges of Clustering?.....	3
CPU.....	3
Memory.....	4
Disk Drives.....	4
First Things First: Why Would I Consider Clustering?.....	4
Benefits of Clustering—Top 3	4
SQL Server Service Packs.....	4
Windows® Service Packs	4
Windows Operating System Crashes.....	4
How Do I Approach Clustering?.....	5
How to install SQL Server 2005 on a cluster.....	5
Installing the Database Engine.....	5
Installing Analysis Services	6
Installing Reporting Services	6
Installing SQL Server Integration Services	6
Installing Tools and Documentation..	6
Installing MSDTC	7
Best Practices	7
OS Windows Server® 2003.....	7
SQL Server 2005	7
Once I Am Finished, How Do I Manage a Clustered Environment?.....	8
Ready to Get Started?— Recap of “How to Get Going”	8



How to Guide: SQL Server 2005 Clustering

Introduction

This *SQL Server 2005 Magazine* white paper is written for a technical audience that needs to understand failover clustering, and that wants to know how SQL Server 2005 is implemented on a failover cluster, how to install and configure SQL Server 2005 for failover clustering, and best practices for SQL Server 2005 clustering.

As SQL Server 2005 environments move from smaller installations to larger mission-critical enterprises, the need for the database environment to be highly available becomes more apparent. SQL Server 2005 has many different mechanisms to achieve high availability. But one of the most commonly used methods to achieve a highly available mission-critical database environment is SQL Server 2005's ability to make use of clustered environments.

Background on SQL Server Clustering: What Is It?

Failover clusters are a Windows solution that allows administrators to create a functional grouping of servers that can act as surrogate hosts for applications running on one or more of the servers in case of unavailability at the server or application level. Simply put, clusters provide high availability for applications running on the servers. This high availability does not guarantee non-stop operations because there is some down-time while the failover occurs; nor does it guarantee the same level of performance after a failover. What a clustered solution guarantees is that cluster-aware applications will be monitored by the clustering software and the application and its resources and then automatically recovered from many failure conditions by running the application on another server.

There are several different resources and components that any administrator needs to understand before considering a clustered solution. These resources and components are considered the foundations of a clustered environment and include the processes needed by the clustering software, the resources needed to store the application code, and the services utilized by the clustering software to manage the clustered solution.

While there are many different resources used by the Windows clustering service, the type of resource that is often the concern of administrators who will install SQL Server 2005 on the cluster is the disk resource group. A resource group (Figure 1) is one or more disks that are grouped together as a functional unit. The disk resource group is considered the failover unit for failover clustering and because the resource group is the failover unit, a separate resource group is needed for each SQL Server 2005 instance being installed in a clustered environment. Each resource group can only be presented to one node at a time and it is this transfer of ownership to a different node that constitutes a failover.

When dealing with SQL Server 2005 and clustering, it is the resource group that determines the failover unit. Because the resource group determines the failover unit, each individual SQL Server 2005 instance must be installed on its own resource group. It is possible to utilize multiple resource groups for a single install but when you do you must set resource dependencies between the two different resource groups so both resource groups will failover as a unit.

Associated with the disk resource group are the outlying dependencies involved in failover clustering. These outlying dependencies include both the network name and the IP addresses used for each resource and will failover with the disk resource group whenever a failover of the node occurs.

Note: Instances that need to create databases or database files on many different disk arrays to spread the I/O over the different arrays can tie all the various arrays back together using resource dependencies. Resource dependencies allow multiple resource groups to failover as a unit if one of the resource groups fails over to another node.

Note: There is a special resource group that is shared by all nodes of a clustered environment. This special resource group, called the Quorum drive, holds a database that is used by the clustering software as a container for the names of the nodes participating in the cluster as well as the system state of each of these nodes.

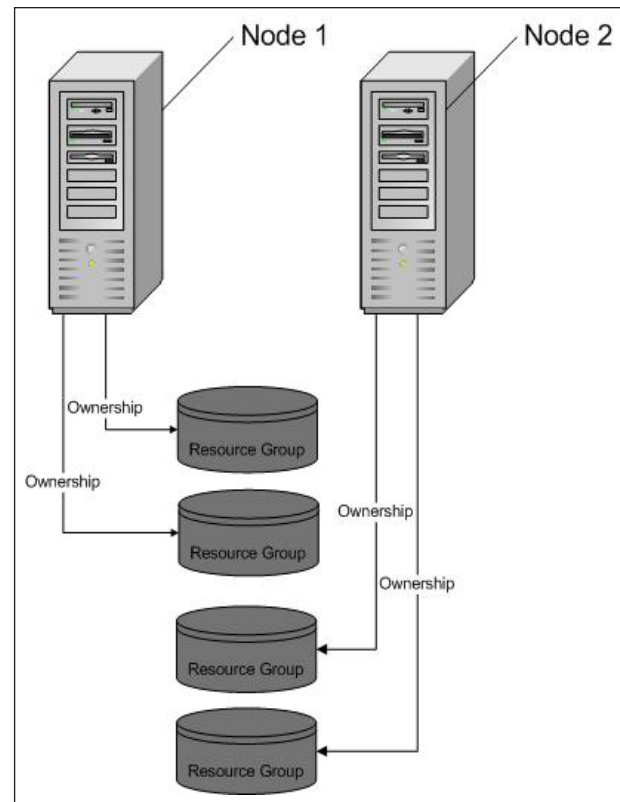


Figure 1 Resource Groups

What types of clusters are there?

Windows Clustering Service offers several different types and modes of clustering: Single-Node clusters, Multi-Node clusters, Majority Node Set clusters, and geographically dispersed clusters. While each of these clusters has its own purposes, the majority of clusters used for SQL Server 2005 consist of either Single-Node or Multi-Node clusters. This white paper will concentrate on those two types of clusters and refer you to the Windows Server 2003 Online help for information on the other two.

Single-Node Clusters

Single-Node clusters (Figure 2) are clustered environments that consist of one actively running server and one or more “inactive” servers. An inactive server is one that does not actively execute applications but is running an OS. This inactive server is “waiting” for the active server to experience a condition that would normally cause an outage so it can host the applications that would normally execute on the active server. At this point the inactive server becomes an active server and all resources needed by the application are put under the ownership of the new active server while all connections that existed on the old active server are reconnected to the new active server.

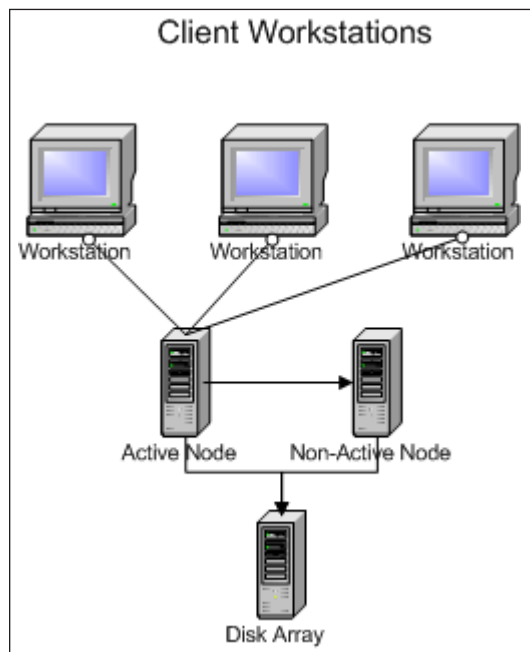


Figure 2 Single-Node Cluster

Multi-Node Clusters

Multi-Node clusters (Figure 3) consist of two to eight active servers running in one cluster group. These active servers are all running their own copies of applications and all doing their own work. Each active server is configured to either act as a host for one or more other active servers in case of a failover or each active server is

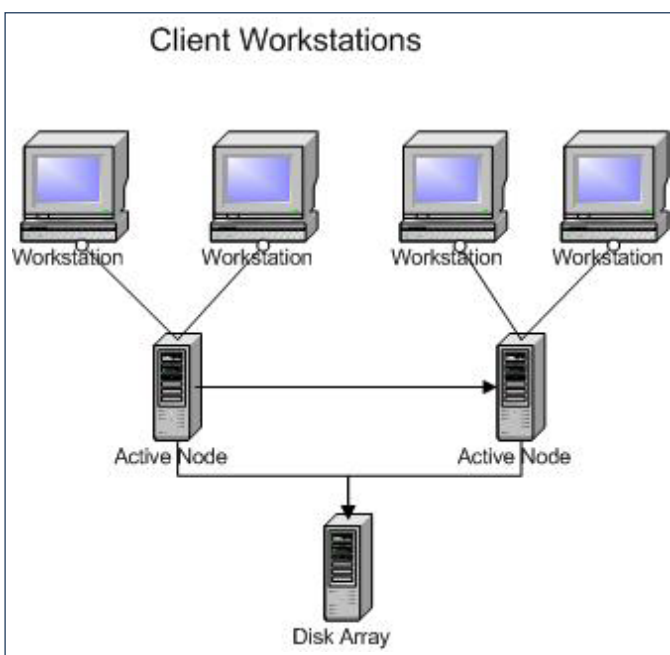


Figure 3 Multi-Node Cluster

configured to use a single active or non-active server as its host in its own failure (Figure 4). Multi-Node clusters allow administrators to use some or all of the resources of each server in the cluster group. Care should be taken in multi-node clusters to account for the performance of the servers if a single server has to act as a host to another server while also hosting its own applications.

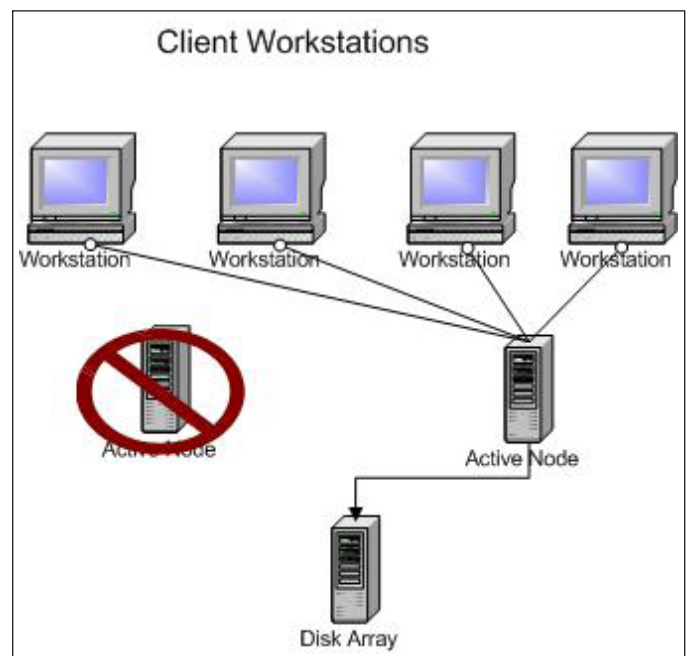


Figure 4 Multi-Node Cluster with Failover

What Are the Top 3 Challenges of Clustering?

When choosing to cluster your SQL Server 2005 database environment, your main challenges will center on resource usage of individual clustered servers. Resource usage is always a concern when managing database environments, but once you start hosting multiple environments on one server—as you would in the case of failover—you will have to be especially careful with your resource usage if you choose to employ a multi-node clustering model.

CPU

CPU resource usage is one of the top three areas you have to plan for when creating clustered environments. The old practice of sizing your CPUs so they are at a utilization rate of 60 percent to 80 percent still holds true, but this utilization rate now has to account for a failover. This means that if you keep your individual nodes at 60 percent to 80 percent utilization they may not be able to handle the workload of failover instances. What does

this mean? To truly size clustered servers for failover, you need to make sure that the total CPU utilization rate after worse-case failover scenarios never goes above 80 percent, even if your normal day-to-day CPU must remain below 40 percent to account for possible failover scenarios.

Memory

As with CPU resource usage, memory is very important to account for in failover clustering. Administrators must account for the memory usage of their applications after a failover. Memory intensive applications such as SQL Server 2005 will use as much memory on their own server as possible and often when failover occurs, the SQL Server 2005 instance that has failed over will have performance issues as it contends with the established instance use of memory.

Note: SQL Server 2005 has been improved over earlier versions of SQL Server 2005 and now has the ability to utilize dynamic memory in clustered environments.

When configuring SQL Server 2005 on a clustered server in a multi-node environment, database administrators are advised to set the **max server memory** configuration setting to leave memory free in case of failovers. It is also recommended that database administrators set the **min server memory** configuration setting to prevent the failed over instance from assuming all of the memory on the new node and affecting the performance of the established instance.

Disk Drives

Often, when creating multi-node clusters, administrators face the issue of not having enough disk drive letters. In a Windows environment, resource groups are given different drive letters on the clustered nodes. As the number of resource groups increases due to having multiple instances of SQL Server 2005 installed or due to the design practice of using multiple filegroups for a database, the 26-drive letter limit is severely restricted as the number of nodes increase in a multi-node environment. Database administrators must work closely with server administrators who often build the Windows clusters before SQL Server 2005 is installed to assure that enough drive letters are available and that the sizes of the drives behind the drive letters are large enough to accommodate their database size.

First Things First: Why Would I Consider Clustering?

The primary purpose of using clustering is for the high availability of your database environments. With today's 24/7 mission-critical databases, having a database unavailable during an outage or during server maintenance is often unacceptable from a business and revenue standpoint. Using clustering in your database environments may help prevent many outages that are caused from server crashes, server maintenance, or even loss of network connectivity to a server.

While clustering does not ensure that there would never be an outage, it does help to reduce the number of outages and allows both database administrators and server administrators options during the outage situations that occur in any server environment.

Benefits of Clustering—Top 3

Clustering offers many benefits to database and server administrators, but the top benefits often cited when using clustering in association with SQL Server 2005 are installation of SQL Server 2005 Service Packs, installation of Windows' Service Packs, and protection from Windows operating system crashes.

SQL Server 2005 Service Packs

As much as the current SQL Server 2005 development team wishes they could achieve service pack installation without downtime, all of the previous and current SQL Server 2005 service packs required database administrators to "bounce" their SQL Server 2005 installations after applying the service pack. When using clustering, database administrators can often achieve service pack installation without downtime by failing over their databases to another node before installing the service pack on the original node. This allows the database to be available on the second node while the instance on the first node is bounced after the service pack installation.

Windows Service Packs

As with SQL Server 2005 service packs, Windows' service packs often require downtime during the process of installing and rebooting after the service pack installation. Having the option of failing over the database to another node will often prevent downtime of the database during Windows' maintenance routines.

Windows Operating System Crashes

The main benefit of having your databases on clustered servers is the prevention of downtime during many of the minor outages that occur on complex server environments. Often small issues happen that lead the operating system to experience a small outage. These outages do not require extensive investigation or rebuild of the server in question, but they are often severe enough to cause applications to go offline. Having clustered servers can prevent many of these applications from going offline as the application can be failed over to the other cluster node without major loss of client connections.

How Do I Approach Clustering?

A clustered environment uses a private network between the nodes to send a “heartbeat” signal to each of the nodes in a server. The purpose of this signal is to check whether each node is alive and capable of operating at both the operating system level and at the application level (SQL Server 2005). At the operating system level this signal serves to keep all the nodes in constant communication and to validate the health of each node of the environment.

Once SQL Server 2005 is installed on the clustered environment, a new signal is sent by the clustering software Service Control Manager every five seconds to each active SQL Server 2005 instance (Figure 5). This signal, called a “LooksAlive,” is very lightweight and does not perform anything more than a simple check to see if the SQL Server 2005 instance is up and running, not if the instance can execute any operations. To determine if the SQL Server 2005 instance can actually execute operations, a deeper “IsAlive” check is performed every 60 seconds by issuing a `SELECT @@SERVERNAME` query to verify a response. If no response is issued, the IsAlive check is executed five more times before the cluster software attempts to start the resource on a different node.

During failover, the clustering software transfers ownership of all resources needed by the SQL Server 2005 instance. Each SQL Server 2005 instance is installed on its own set of disks called a resource group and it is this resource group that has its ownership transferred to the second node. After the resource group is transferred, the SQL Server 2005 instance undergoes the recovery process and recovers the system databases to bring the SQL Server 2005 service online. The service is considered to be online once the **master** database is online. Once the **master** database is online the service starts to recover the user databases.

Client applications will undergo reconnections based on the configuration of application. When applications connect to a SQL Server 2005 instance they do so through virtual and not physical IP addresses. The actual location of the virtual IP address is controlled by the clustering software so the applications never have to know which node the instance actually resides on. You can utilize the

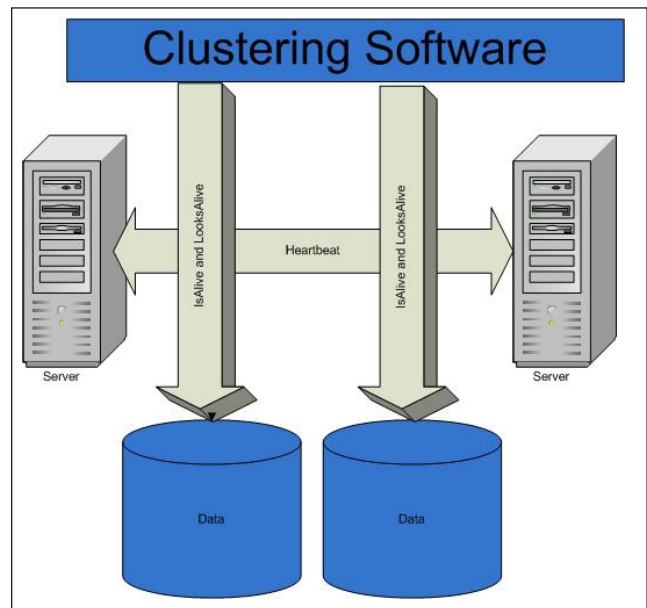


Figure 5 Clustered Environment Health Checks

physical IP address to connect to the SQL Server 2005 service but if you do so, you will not be able to connect after a failover if the node hosting the SQL Server 2005 service is the one that undergoes the failure.

How to install SQL Server 2005 on a cluster

While installing SQL Server 2005 on a non-clustered environment is very straight forward, installing SQL Server 2005 on a clustered environment involves a few more steps. SQL Server 2005 failover clustering requires the Enterprise Edition (supports up to an 8-node cluster), Developer Edition (supports up to an 8-node cluster), or Standard Edition (supports up to a 2-node cluster) and an edition of Windows that supports clustering: Windows Server 2003 Enterprise, Windows 2003 Datacenter Edition, Windows 2000 Advanced Server, or Windows 2000 Datacenter Edition.

Administrators installing SQL Server 2005 on a cluster will need to work with their server administrators to configure the correct number of resource groups, drive letters, drive sizes, and the IP addresses needed for the SQL Server 2005 installation. Database administrators should discuss the components they need to install and how those components behave in a clustered environment.

Installing the Database Engine

Database administrators installing SQL Server 2005 on a failover cluster will be pleased to know that SQL Server 2005 is cluster aware during the install process. This means that the installation wizard will automatically install the SQL Server 2005 instance with the correct configuration on both nodes of a clustered environment.

For those installed in a multi-node environment, you will still need to install the active instance on node 1 and then the active instance on each of the other nodes.

Note: You can only have 1 default instance in a clustered environment which means that you will need to install named instances on all but one node of the clustered environment.

Note: Please review the following SQL Server 2005 Books Online article for installing a SQL Server 2005 database engine instance: *How to: Create a New SQL Server 2005 Failover Cluster (Setup)* at <http://msdn2.microsoft.com/en-us/library/ms179530.aspx>.

Installing Analysis Services

As with the database engine, the SQL Server 2005 installation wizard will install an Analysis Services instance on both nodes in a clustered environment. Please review the reference material found in the SQL Server 2005 Books Online Article: *How to: Install Analysis Services on a Failover Cluster* at <http://msdn2.microsoft.com/en-us/library/ms143511.aspx>.

Installing Reporting Services

Unfortunately, Reporting Services is not cluster-aware and will have to be installed as stand-alone components on all nodes in the clustered environments. Reporting Services can make use of clustered databases. But when a node fails over the connections to any Reporting Services, instances on the failed node will not be automatically reconnected to the new node. Connections will have to attempt to connect to the new node using a new IP address.

Installing SQL Server Integration Services

To install SSIS, you must install SSIS on all active nodes and then make the SSIS Service a clustered resource. To make SSIS a clustered resource, perform the following tasks:

1. Open the **Cluster Administrator**
2. On the **File** menu, point to **New** and then click **Resource**
3. On the **New Resource** page of the **Resource Wizard**, type a Name and choose "Generic Service" as the

Service Type. Change the value of **Group** to **SQL group**. Click **Next**.

4. On the **Possible Owners** page, add or remove the nodes of the cluster as the possible owners of the resource. Click **Next**.
5. To add dependencies on the **Dependencies** page, select a resource under **Available resources**, and then click **Add**. In case of a failover, both SQL Server 2005 and the shared disk that stores Integration Services packages should come back online before Integration Services is brought online. After you have selected the dependencies, click **Next**.
6. On the **Generic Service Parameters** page, enter **MsDtsServer** as the name of the service. Click **Next**.
7. On the **Registry Replication** page, click **Add** to add the registry key that identifies the configuration file for the Integration Services service.
 - a. This file needs to be located on a shared disk that is in the same group for the Integration Services service, and fails over to the next node along with for the Integration Services service.
 - b. In the **Registry Key** dialog box, type `SOFTWARE\Microsoft\MSDTS\ServiceConfigFile`. Click **OK**, and then click **Finish**. The Integration Services service has now been added as a clustered service.
8. Locate the configuration file at `%ProgramFiles%\Microsoft SQL Server 2005\90\DTS\Binn\MsDtsSrvr.ini.xml`. Copy it to the shared disk.
9. Create a new folder named **Packages** on the shared disk. Grant **List Folders and Write** permissions on the new folder to the built-in Users group.
10. Open the configuration file from the shared disk in a text or XML editor. Change the value of the **ServerName** element to the name of the virtual SQL Server 2005 which is in the same group.
11. Change the value of the **StorePath** element to the fully-qualified path of the Packages folder created on the shared disk in a previous step: `??:\Packages`.
12. Update the value of `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSDTS\ServiceConfigFile` in the Registry to the fully-qualified path and filename of the service configuration file on the shared disk
13. In the **Cluster Administrator**, select the Integration Services service, right-click, and select **Bring Online** from the popup menu. The Integration Services service is now online as a clustered service

Installing Tools and Documentation

SQL Server 2005 Setup no longer installs the tools and documentation components of SQL Server 2005 on all the nodes of a cluster. The wizard will only install these components on the node that the Setup routine is initiated from. If tools and documentation is required on the

other nodes, you will need to install these components manually after finishing the setup of the cluster.

Installing MSDTC

Microsoft Distributed Transaction Coordinator is often used during SQL Server 2005 operations to handle distributed transactions. Database and server administrators should understand that MSDTC does not install on a cluster without a few extra sets that must be followed. MSDTC now has the ability to be installed on all the servers of a clustered environment when using Windows 2003, in the past you had to install MSDTC on each server separately. Windows 2003 gives you two different options when installing MSDTC: Use the cluster administrator to install MSDTC or use Cluster.exe to install MSDTC.

Note: In place of walking you through the steps needed to install MSDTC on a Windows 2000 or Windows 2003 cluster, I am going to refer you to the following two Microsoft support articles:

- **How to configure Microsoft Distributed Transaction Coordinator on a Windows Server 2003 cluster:** <http://support.microsoft.com/kb/301600>
- **How to configure MSDTC in a Windows 2000 cluster environment:** <http://support.microsoft.com/kb/q290624/>

Best Practices

OS Windows Server 2003

- You should spend more time designing and planning your clustered environment than actually creating the clustered environment.
- All hardware used for the clustered environment must be on the Windows 2003 Clustering HCL as a unit and not as individual pieces.
- Each cluster node should have identical hardware
- If creating a clustered environment with more than 2 nodes, you should consider making one of the nodes a passive node and configure it as the primary failover node.
- Determine all virtual names and IP addresses during the planning phase
- Be sure to account for future additional space during the configurations of the drives and drive letters

- Use a standard drive letter for the local drives and quorum drives across all clustered nodes in your environment
- It is recommended that you utilize two separate (virtual or physical) networks in a clustered environment. One network for the public and one for the private. Be sure to use separate network cards for the Public and Private networks and not a single multi-port network card.
- Use static IP addresses for both the Public and Private networks with the Private network using a private class A, B, or C address.
- Make sure the shared disk array is configured properly, running, and can be seen by all nodes before installing your cluster.
- Make sure your Quorum drive is fault tolerant.
- Create unique virtual server names of 15 characters or less.
- Be sure to disallow automatic fail-back to avoid the “ping-pong” effect resource bouncing back and forth when one node constantly fails and comes back up.

SQL Server 2005

- Microsoft recommends installing Analysis Services clustered installations in a single group with its own disk and IP resources and not on a resource group that has the database engine installed on it.
- While the Enterprise Edition of SQL Server 2005 supports 25 instances on a clustered node, Standard Edition only supports 16, you will need to understand that each instance needs its own resource group and you will need to utilize the Majority Node Set model. It is usually not best practice to install more than a few instances on a clustered node in order to achieve scalability and performance goals.
- SQL Server 2005 requires the .NET Framework on all nodes of the cluster and while the installation wizard will install the .NET Framework on all nodes, it is usually recommended that you install the .NET Framework on each node before attempting the actual installation process in order to speed up the install process.
- It is usually recommended that you create four different domain groups when creating a cluster. One for each of the following: SQL Server 2005 service, SQL Server 2005 Agent service, Full-Text Search service, and Analysis Services service.

- Clustered SQL Server 2005 nodes should not have other applications installed on them
- Since you can only have one default instance in a clustered environment, it is better to simply name all instances in your environment for standardization purposes.
- Do not delete or rename the default cluster group, or remove any of the resources from this group and do not delete or rename any resources from the SQL Server 2005 resource group.
- If using replication in a clustered environment, place the snapshot folder on a network share that all nodes can read and write to.
- Be sure to test the failover of all nodes before going into production

Once I Am Finished, How Do I Manage a Clustered Environment?

For a database administrator, managing SQL Server 2005 on a clustered environment does not imply that you have to do anything different than you would if the database was not on a clustered environment. The differences in running a SQL Server 2005 instance on a clustered environment come mainly during the installation phase and not during the day-to-day management of SQL Server 2005. If you are running a multi-node environment, you do have to worry about if your resource groups have failed over or not, as having multiple instances on one node may put a performance strain on that node.

Another issue that you will need to worry about, especially when you are running a single-node environment is your performance monitoring. Be sure that your monitoring solution will understand the failover process and will continue to provide monitoring after a failover occurs. Many monitoring solutions do not failover to the second node and will continue to provide metrics for the node that does not have SQL Server 2005 currently executing on it which may invalidate your trending values.

Ready to Get Started?— Recap of “How to Get Going”

Once you have a clustered environment ready for SQL Server 2005, you will only need to start the standard installation process. After installing the database engine and Analysis Services, you will need to install SSIS on both nodes and configure it as a clustered resource. You will also need to install Reporting Services, Tools, and Documentation on all nodes in order to use them after failover. Once this is done, make sure your clients are using the virtual IP addresses and names and you are ready to assume your day-to-day responsibilities against the clustered instances. Be sure to have your server group put you on the notification list in case of node failover so you can help them take action to prevent performance issues after failover.

Administering a clustered instance is not all that different from administering standalone instances. Make sure you utilize the virtual IP address and names in your administration scripts and you will be amazed at how soon you forget that the instance is on a clustered node.

Randy Dyess, Solid Quality Learning Mentor and Program Manager: Strategic Initiatives, has a variety of experiences dealing with SQL Server 2005 over the past nine years and has worked with environments with Terabytes of data and environments that had over 1,000 databases with only a few megabytes of data in each database. Currently, Randy is the founder and owner of Dyess Consulting Inc. a SQL Server 2005 mentoring and training consulting firm which specializes in training and mentoring in Transact-SQL and SQL Server 2005 performance tuning and database security. Randy is the author of *Transact-SQL Language Reference Guide* and numerous magazine and newsletter articles pertaining to SQL Server 2005 security and optimization issues and has spoken at various international and national conferences.

Dell and PowerEdge are trademarks of Dell Inc. Microsoft, SQL Server, and Windows are registered trademarks of Microsoft Corporation. Other trademarks and trade names may be used in this document to refer to either the entities claiming the marks and names or their products. Dell disclaims proprietary interest in the marks and names of others.