



[TechNet Home](#) > [Product & Technologies](#) > [SQL Server TechCenter Home](#) > [SQL Server 2000](#) > [Maintain](#)

SQL Server 2000 Failover Clustering

Published: June 18, 2002 | Updated: October 14, 2004

Note: You can also [read a SQL Server 2005 version](#) of this paper.

Abstract: Users demand not only performance and scalability, but availability of the applications they use. Achieving high availability requires a whole solution, part of which includes the SQL Server 2000 Server failover clustering. This white paper will explain what failover clustering is: how it works, considerations to take into account when designing your solution, how to implement and administer your solution, and how to troubleshoot a failover cluster. Also included are some useful worksheets and checklists to assist you during the installation, as well as links to additional information where necessary.



On This Page

- ↓ [Introduction](#)
- ↓ [Enhancements to Failover Clustering](#)
- ↓ [What Is Windows Clustering?](#)
- ↓ [What is SQL Server 2000 Failover Clustering?](#)
- ↓ [Configuring SQL Server 2000 Failover Clustering](#)
- ↓ [Implementing SQL Server 2000 Failover Clustering](#)
- ↓ [Verifying Your Failover Cluster Installation](#)
- ↓ [Maintaining a SQL Server 2000 Failover Cluster](#)
- ↓ [Troubleshooting SQL Server 2000 Failover Clusters](#)
- ↓ [Conclusion](#)
- ↓ [Appendix A –Additional Information](#)
- ↓ [Appendix B – Step-By-Step Installation Instructions for a New Virtual Server](#)
- ↓ [Appendix C - Configuration Worksheets](#)
- ↓ [Appendix D – Pre- and Post-Installation Checklists](#)

Introduction

Continuous uptime in a production environment, whether it is a database powering a mission-critical client/server application or an e-commerce Web site, is becoming a common business requirement. This paper describes Microsoft® SQL Server™ 2000 failover clustering, one method of creating high availability. Failover clustering is available only in SQL Server 2000 Enterprise Edition.

Failover clustering is a process in which the operating system and SQL Server 2000 work together to provide availability in the event of an application failure, hardware failure, or operating-system error. Failover clustering provides hardware redundancy through a configuration in which mission critical resources are transferred from a failing machine to an equally configured server automatically. Failover clustering also allows system maintenance to be performed on a computer while another node does the work. This benefit can also ensure that system downtime due to normal maintenance is minimized. For more information about optimizing your database and tips on how to avoid performance issues that can lead to unavailability, see Chapter 33, "The Data Tier: An Approach to Database Optimization," in the *Microsoft SQL Server 2000 Resource Kit*.

The goal of failover clustering is to provide high availability for an overall scale-up and scale-out solution that accommodates backups, redundancy, and performance. If software and/or hardware problems occur, failover clustering combined with other high availability methods (such as SQL Server 2000 log shipping) can enable a production environment to be up and running in a short amount of time.

However, failover clustering is not a load balancing solution and it cannot protect your system against external threats, catastrophic software failures to all nodes of the cluster, single points of failure (such as non-redundant hardware), or natural disasters. For more information about SQL Server 2000 high availability, see Chapter 16,

"Five Nines: The Ultimate in High Availability," in the *Microsoft SQL Server 2000 Resource Kit*.

[↑ Top of page](#)

Enhancements to Failover Clustering

Microsoft SQL Server 2000 Enterprise Edition failover clustering offers improvements over the clustering functionality provided in SQL Server version 7.0 Enterprise Edition. Some of the enhancements to the clustering implementation in SQL Server 2000 include:

- Installing and uninstalling a SQL Server 2000 failover cluster are both now done via the SQL Server 2000 setup program, and not through the combination of setting up your database server and then a Wizard. Installation and clustering are done in one process. SQL Server 2000 failover clustering is a permanent option, and the only way to remove it is to uninstall the clustered instance of SQL Server.
- SQL Server 2000 supports multiple instances, allowing simultaneous support of up to 16 instances of SQL Server.
- SQL Server 2000 has extensive support for recovering from a failure of a server node in the cluster, including a one-node cluster. If a node fails it can be removed, reinstalled, and rejoined to the cluster while all other nodes continue to function properly. It is then a simple operation with SQL Server 2000 Setup to add the new server back into the virtual server definition.
- SQL Server 2000 running on Microsoft Windows® 2000 Datacenter Server supports up to four server nodes in a cluster.
- All nodes now have local copies of the SQL Server tools (including performance counters) as well as the executables so in the event of a failover; you can administer the server from a remote system or the clustered node itself.
- SQL Server 2000 failover clustering supports Microsoft Search Services.
- SQL Server 2000 failover cluster configurations can be updated by rerunning the setup program.
- SQL Server 2000 supports multiple network addresses. This enables SQL Server 2000 to listen on multiple IP addresses on different subnets.
- Database administrators can now use SQL Server Service Manager or SQL Server Enterprise Manager to start and stop SQL Server without having to use Cluster Administrator to start and stop SQL Server services.
- Service packs are applied directly to the SQL Server 2000 virtual server. With SQL Server 7.0, you had to uncluster the server prior to applying a service pack.
- SQL Server 2000 is now a fully cluster aware application. This allows SQL Server 2000 to interact with the Cluster service, and it provides some benefits such as preventing the creation of databases on invalid logical drives.

[↑ Top of page](#)

What Is Windows Clustering?

Microsoft SQL Server 2000 failover clustering is integrated with Windows Clustering. There are two main types of clusters in a Windows environment:

- Server cluster

SQL Server 2000 failover clustering is built on top of a Windows 2000 Advanced or Datacenter server cluster. A Windows 2000 server cluster provides high availability, scalability, and manageability for resources and applications by clustering as many as four servers to maintain client access to applications and server resources during unplanned outages due to hardware failures, natural and man-made disasters, software failure, and so on. Unlike the behavior of a Network Load Balancing cluster, when a server, resource, or cluster-aware application within the cluster becomes unavailable, it will be transferred to another server that is available.

- Network Load Balancing cluster

A Network Load Balancing cluster provides high availability and scalability for TCP/IP-based services, including Web servers, FTP servers, other mission-critical servers, and COM+ applications. In a Network Load Balancing scenario, multiple servers run independently, and do not share any resources. Client requests are distributed among the servers, and in the event of a server failure, a Network Load Balancing cluster detects the problem and the load is distributed to another server. SQL Server 2000 failover clustering does not fall into this

category, but may be part of an overall architecture in which a Web farm using a Network Load Balancing cluster connects to a failover cluster. Because you employ a Network Load Balancing cluster due to an application requirement, you need to consider Network Load Balancing during the application planning and configuration stage.

Hardware for Windows Clustering

The following is a list of hardware components used in Windows Clustering, which is a feature of Microsoft Windows 2000 Advanced Server, Windows 2000 Datacenter Server, Windows 2003 Enterprise Edition, Windows 2003 Datacenter Edition and Microsoft Cluster Service (MSCS), which is a feature of Microsoft Windows NT® 4.0, Enterprise Edition:

- Cluster Nodes

A node is a server within the cluster. Windows NT Server 4.0, Enterprise Edition and Windows 2000 Advanced Server and Windows 2003 Advanced Server both support two-node clustering, and Windows 2000 Datacenter Server supports up to four-node clustering and Windows 2003 supports up to eight node clustering however you are limited to four nodes if SQL Server 2000 clustering is to be used. For more information, see Knowledge Base article "811054 PRB: Virtual SQL Server 2000 Installation Fails on Cluster That Has Eight Nodes" <http://support.microsoft.com/?id=811054>.

- Heartbeat

The heartbeat is a private network set up between the nodes of the cluster that checks to see whether a server is up and running. This occurs at regular intervals known as time slices. If the heartbeat is not functioning, a failover is initiated, and another node in the cluster will take over the services.

- External Networking

In addition to the heartbeat private network, at least one public network must be enabled so external connections can be made to the cluster.

- Shared Cluster Disk Array

The shared disk array is a collection of physical disks (SCSI RAID or FibreChannel) that is accessed by the cluster. Windows Clustering supports shared nothing disk arrays. A shared nothing disk array is a setup in which only one node can own a given resource at any given moment. All other nodes are denied access until they own the resource. This protects the data from being overwritten data when two computers have access to the same drives concurrently.

- Quorum Drive

The quorum drive is a logical drive designated on the shared disk array for Windows Clustering. This continuously updated drive contains information about the state of the cluster. If this drive becomes corrupt or damaged, the cluster installation also becomes corrupt or damaged.

Operating System

The following is a list of components, also known as cluster resources, which are exposed at the operating-system level:

- Cluster Name

The name that all Windows NT or Windows 2000 external connections use to refer to the cluster itself, but not the SQL Server virtual server; the individual nodes are never referenced.

- Cluster IP Address

The IP address that all external connections use to reach the failover cluster itself, not the SQL Server virtual server.

- Cluster Administrator Account

This account is used to administer and own the failover cluster. A Cluster Administrator account must be created at the domain level and must be an administrator of all nodes in the cluster.

- Cluster Resource Types

Cluster resources include any services, software, or hardware that can be configured within a cluster. These include: DHCP, File Share, Generic Application, Generic Service, Internet Protocol, Network Name, Physical Disk, Print Spooler, and WINS.

- Cluster Group

A cluster group is a collection of logically grouped cluster resources, and may contain cluster-aware application services such as SQL Server 2000. Conceptually, a cluster group is a folder on your hard drive that contains related information.

Virtual Server

Understanding the concept of a virtual server is a key to understanding failover clustering. To a client or application, a virtual server is the server name or IP address (es) used for access. The connection from the client to the virtual server does not need to know which node within a cluster is currently hosting the virtual server. A clustered SQL Server is known as a SQL Server virtual server.

[↑ Top of page](#)

What is SQL Server 2000 Failover Clustering?

SQL Server 2000 is built on top of Windows Clustering or MSCS because it is a cluster-aware application. In Figure 1, the virtual server of SQL Server 2000 sits on top of the existing MSCS installation.

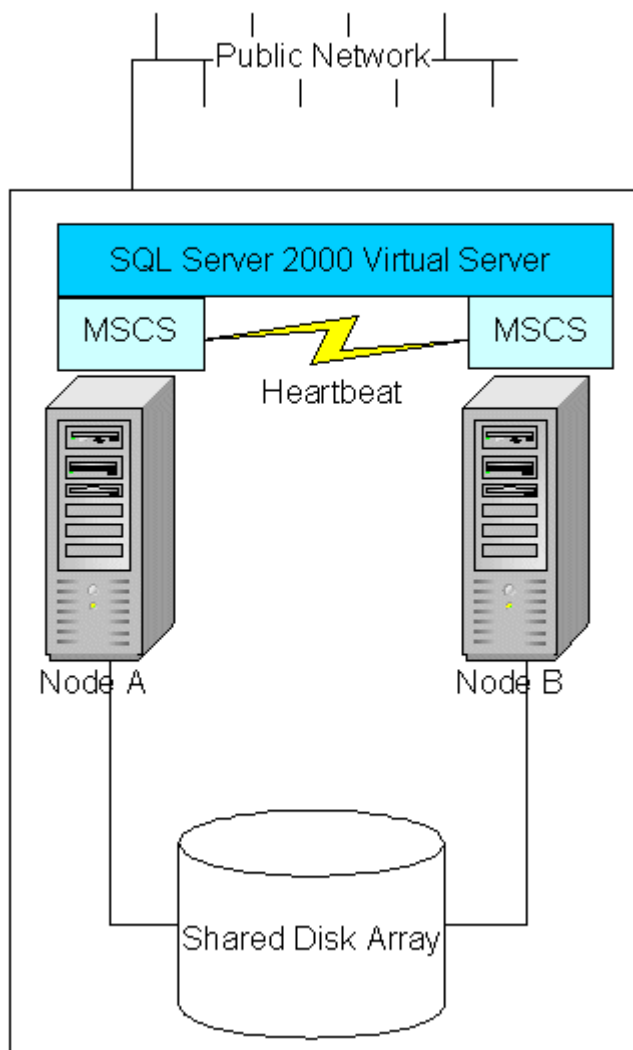


Figure 1: SQL Server 2000 virtual server illustration. This example is comprised of two server nodes, and one SQL Server 2000 virtual server.

SQL Server Virtual Server Components

An instance is an installation of SQL Server that is completely separate from any other, with a few underlying shared components that affect how SQL Server 2000 works in a clustered environment. A SQL Server virtual server is an instance of SQL Server that has been clustered. The following resources make up each virtual server:

- SQL Server Network Name

This is the name that users and applications will use to connect to SQL Server.

- SQL Server IP Address

The TCP/IP address that users and applications will use to connect to SQL Server. This is different from the Cluster IP Address

- SQL Server

This controls this instance of the SQL Server 2000 service.

- SQL Server Agent

This controls this instance of the SQL Server Agent service.

- SQL Server 2000 Full-text

Each virtual server also has one full-text resource, unlike the SQL Server and SQL Server Agent resources; each instance refers to the shared Microsoft Search service. In the event of a failover, it is not the same as other services; only the data files are failed over, not the service.

- Microsoft Distributed Transaction Coordinator (MS DTC)

Some installations of SQL Server utilize MS DTC. If this is the case for your installation, MS DTC is shared for all instances in the cluster.

- SQL Server Virtual Server Administrator Account

This is the SQL Server service account. This account may be the same as the Cluster Administrator account described earlier. The service account must also have administrator permissions on all nodes if you are using Windows NT 4.0 Enterprise Edition, but not if you are using Windows 2000. For more information about creating this account, see "Setting up Windows Services Accounts" in SQL Server 2000 Books Online.

As noted in the section "Enhancements to Failover Clustering", SQL Server 2000 supports multiple instances per server — one default instance, and up to 15 named instances, or 16 named instances. SQL Server can be installed either as a default instance or as a named instance. A SQL Server 2000 virtual server can also have local named instances or a local SQL Server 7.0 default instance, but these will not appear visible to Windows Clustering. These are instances local to the server.

Important: An instance of SQL Server 2000 cannot be run on a SQL Server 6.5 or SQL Server 7.0 cluster.

With instances, come two new concepts for failover clustering:

- Single instance cluster: replaces an active/passive cluster. A single instance cluster means there is one SQL Server 2000 virtual server installed.
- Multiple instance cluster: replaces an active/active cluster. A multiple instance cluster is one in which there is more than one SQL Server 2000 virtual server installed. Because of the way the implementation of clustering is different with SQL Server 2000, using the active/active terminology does not really apply.

Single-Instance Cluster

A single-instance cluster has only one active instance of SQL Server owned by a single server node, and all other nodes of the cluster are in a wait state. Another node is enabled in the event of a failure on the active node, or during a manual failover for maintenance.

Multiple-Instance Cluster

A multiple-instance cluster has up to four server nodes and supports up to 16 instances (1 default, 15 named or 16 named). Each SQL Server 2000 virtual server requires its own disk resources that cannot be used by other instances. These disk resources are the logical drive names (for example, drive F:\) used by SQL Server on which to store data and log files. Separate physical disk sets are needed to make up the logical drive, unless your disk subsystem supports multiple logical drives on one physical drive set. SQL Server in a clustered environment also behaves differently from a stand-alone named instance in relation to IP ports. During the installation process, a dynamic port that may be something other than 1433 is configured, and that port number is reserved for the instance. In a failover cluster, multiple instances can be configured to share the same port, such as 1433, because the failover cluster listens only to the IP address assigned to the SQL Server virtual server, and is not limited to a 1:1 ratio. However, for security and potentially increased availability, you may want to assign each virtual server to its own unique port of your choice or leave it as it was configured during installation.

How Failover Clustering Works

The clustered nodes use the heartbeat to check whether each node is alive, at both the operating system and SQL Server level. At the operating system level, the nodes in the cluster compete for the resources of the cluster. The primary node reserves the resource every 3 seconds, and the competing node every 5 seconds. The process lasts for 25 seconds and then starts over again. For example, if the node owning the instance fails due to a problem (network, disk, and so on), at second 19, the competing node detects it at the 20-second mark, and if it is determined that the primary node no longer has control, the competing node takes over the resource.

From a SQL Server perspective, the node hosting the SQL Server resource does a looks-alive check every 5 seconds. This is a lightweight check to see whether the service is running and may succeed even if the instance of SQL Server is not operational. The IsAlive check is more thorough and involves running a `SELECT @@SERVERNAME` Transact SQL query against the server to determine whether the server itself is available to respond to requests; it does not guarantee that the user databases are up. If this query fails, the IsAlive check retries five times and then attempts to reconnect to the instance of SQL Server. If all five retries fail, the SQL Server resource fails. Depending on the failover threshold configuration of the SQL Server resource, Windows Clustering will attempt to either restart the resource on the same node or fail over to another available node. The execution of the query tolerates a few errors, such as licensing issues or having a paused instance of SQL Server, but ultimately fails if its threshold is exceeded.

During the fail over from one node to another, Windows clustering starts the SQL Server service for that instance on the new node, and goes through the recovery process to start the databases. The fail over of the SQL Server virtual server will take a short time (probably seconds). After the service is started and the **master** database is online, the SQL Server resource is considered to be up. Now the user databases will go through the normal recovery process, which means that any completed transactions in the transaction log are rolled forward, and any incomplete transactions are rolled back. The length of the recovery process depends on how much activity must be rolled forward or rolled back upon startup. Set the recovery interval of the server to a low number to avoid long recovery times and to speed up the failover process.

Client Connections and SQL Server 2000 Virtual Servers

End users and applications access a SQL Server 2000 virtual server with the SQL Server Network Name or IP address of the SQL Server 2000 virtual server. The Cluster name, the Cluster IP address, or even the individual node names are not used by the connections. From a client or application perspective, it does not need to worry about which node owns the resources, because connecting to the SQL Server 2000 virtual server appears as a normal SQL Server. During the failover process, any active connections are broken. For Web browser users, a simple refresh of the Web page should create a new database connection. In a more traditional client/server application, or one that relies heavily on a middle tier, application designers may want to consider checking to see whether the connection exists, and if not, reconnect. Therefore, whatever the user was working on when the server went down may not be completed, unless the transaction completes before the server goes down or the transaction is handled within the application.

For more information, see the Knowledge Base article "273673 – Virtual Server Client Connections Must be Controlled by Clients" at: <http://support.microsoft.com/default.aspx?scid=kb:en-us:273673&sd=tech>

[↑ Top of page](#)

Configuring SQL Server 2000 Failover Clustering

Perhaps the most important aspect of a successful SQL Server 2000 failover cluster installation is ensuring that the right hardware and software are correctly deployed for the application designed to run on the failover cluster. The hardware should be high performance, and scale along with the specific needs of the application(s) accessing SQL Server.

Designing Your Application for a Failover Cluster

Before you get to design the hardware, you must take into account the behavior during a potential failover. A few application design considerations must be taken into account when working with a failover cluster.

- Make all transactions as small as possible, and commit in logical units of work. Since a virtual server goes through the startup process, which includes going through the transaction log for each database and rolling transactions back or forward, the larger the transaction size along with a larger volume of transactions could result in a slower failover time.
- An application is considered cluster-aware if it uses the Windows Clustering Server Cluster APIs.
- Set timeout values in the application effectively to gracefully close connections or do some other appropriate response, such as a friendly message, so that the user experience is a positive one. The end user should never have to worry about the database.

- In conjunction with the previous bullet point, use retry logic to reconnect to the database if the connection is broken. Some applications, such as Microsoft BizTalk Server, have retry logic as part of the programming model. However, if no such provision exists, a custom solution may need to be devised, such as using some sort of middleware.

Administrator Accounts and SQL Server 2000 Failover Clustering

There are a few Windows-level accounts that need to be configured prior to installing both the server cluster and the SQL Server 2000 virtual server.

- An account must be created for the administration and ownership of the server cluster. It must be a valid domain administrator account. This account is also used during the installation of the SQL Server 2000 virtual server.
- At least one account must be created that will administer the SQL Server as well as the SQL Server Agent. This can be two separate accounts, and does not need to be a domain administrator, but a valid domain account. If desired, it can be the same as the account listed in the prior bullet point, but it is helpful to keep the accounts separate.

Although the account is automatically assigned the proper privileges during the installation process, if the account is changed, it must have the following (or the Administrator group must have these):
It must be a member of the local Administrators.

It must be granted the policies of "Act as part of the operating system", "Log on as a service", and "Replace a process level token."

The service account for the Cluster service must have the right to log in to SQL Server. If you accept the default, the account [NT Authority\System] must have login rights to SQL Server so that the SQL Server resource DLL can run the IsAlive query against SQL Server. If corporate policy requires restricting access please follow the instructions documented in Knowledge Base article, "263712 INF: How to impede Windows NT administrators from administering a clustered instance of SQL Server", <http://support.microsoft.com/?id=263712> and "291255 BUG: IsAlive Check Does Not Run Under the Context of the BUILTIN\Administrators Account", <http://support.microsoft.com/?id=291255>.

Note: Keep in mind that any corporate policy that requires the changing of an account's password (such as having to change it every 90 days) will potentially affect your virtual server's availability because you will need to reconfigure each SQL Server 2000 virtual server, including stopping and restarting it for the change to take affect. This must be taken into account when planning the amount of availability your environment needs, and balancing it with corporate security.

Important: Use SQL Server Enterprise Manager if you need to change the accounts associated with the SQL Server virtual server (SQL Server or SQL Server Agent). This will change the service password on all the nodes and grant the necessary permissions to the chosen user account. If SQL Server Enterprise Manager is not used to change passwords, and the Windows-based Services tool is used to modify the underlying service, you may not be able to start SQL Server after a shutdown or a failover, and things such as full-text search may not function properly.

Security

If advanced security, such as Kerberos, SSL, or IPSEC, is part of your overall solution, consider the following when planning your failover cluster implementation:

- Kerberos can be used to authenticate a connection to a cluster virtual server; if unable to connect using Kerberos authentication the clients will try with NTLM authentication.
- If a SSL certificate with the same name as your SQL Server virtual server is installed, the SQL Server instance may not start. For more information, see the Knowledge Base article 283794 "Problems Using Certificate with Virtual Name in Clustered SQL Servers" at <http://support.microsoft.com/default.aspx?scid=kb;en-us;283794>.

Software Requirements

SQL Server 2000 failover clustering requires SQL Server 2000 Enterprise Edition and one of the following operating systems:

- Microsoft Windows NT Server 4.0, Enterprise Edition (with a minimum of Service Pack 5)
- Microsoft Windows 2000 Advanced Server
- Microsoft Windows 2000 Datacenter Server
- Microsoft Windows 2003 Enterprise Edition, SQL Server SP3 or SP3a required

- Microsoft Windows 2003 Datacenter Edition Server, SQL Server SP3 or SP3a required

Hardware Requirements

A SQL Server 2000 virtual server should not only be an instance of SQL Server that is highly available, but one that is highly performing and scalable. Two main factors determine hardware needs:

- What is the current workload of the application or Web site, and what is the projected workload in six months, a year, or even two years from now?

This is information that most people do not have prior to implementing a solution. Having benchmarks on how an application or Web site performs is critical in determining which operating system and what hardware to buy. The best way to evaluate an application is in a lab environment. Using tools such as System Monitor (Performance Monitor in Windows NT 4.0) can also establish performance trends. Without a baseline, or some sort of performance documentation, it will be difficult to determine exact needs. Additionally, take into account, any application issues affecting performance, in either current production versions or updates planned.

- How much money is budgeted to the project?

Although money should not be a barrier to availability, reality dictates that there is a budget to take into account. Prior to purchasing your cluster solution, assess your hardware needs with the following:

- How long do you plan to keep the servers in service?
- Do you have the proper amount of disk space to last you for that period?
- Do you have the proper memory and CPU capacity for this duration?

This kind of planning may prevent you from outgrowing your hardware in terms of performance and capacity sooner than expected. As a result, your solution will be more available because you will not need to upgrade as often.

Important: Configure all nodes in a failover cluster so that they are at least equal to one another. However, if you plan to configure one node so that it contains a greater number of virtual servers than the others, configure that node so that it can handle the capacity of all virtual servers that it will be expected to host. Underpowering a node may affect availability.

For a primary high availability solution, consider failover clustering. However, if you cannot afford to purchase the required complete clustered solution from the Microsoft Hardware Compatibility List (HCL), consider another high availability options such as log shipping, which requires HCL-compatible equipment, but does not require you to buy a complete solution or specialized hardware for a Windows cluster.

Hardware Compatibility List

Before deciding on all final hardware, consult the Windows Server Catalog. The Windows Server Catalog replaces the Cluster Hardware Compatibility List (HCL) that is still accessible at the following Microsoft Web site:

<http://www.microsoft.com/whdc/hcl/search.mspix>

The complete hardware solution must appear under the server configurations found under the "Cluster" category. Buying individual components will not create a supported solution, even if the hardware purchased is on the HCL. If the solution does not appear on the HCL, the cluster configuration is not supported.

Microsoft server clusters are only supported on cluster solutions that are listed in the Windows Server Catalog under Cluster Solutions. To view the Windows Server Catalog, visit the following Microsoft Web site:

<http://www.microsoft.com/windows/catalog/server/default.aspx?xslt=category&subid=22&pgn=904c28be-5a41-4db0-9c12-032dcb893c8b>

Note: The term "server clusters" means computers that run the Microsoft Cluster Service, not the Network Load Balancing or the Windows Load Balancing Service. Supported SQL Server failover clustering installations must also follow the Microsoft support policy for server clusters, and the Windows Server Catalog/Hardware Compatibility List. For any update to the SQL Server Cluster Support policy see Knowledge Base article, "327518 INF: The Microsoft Support Policy for a SQL Server Failover Cluster", <http://support.microsoft.com/?id=327518>

Processors

Depending on the operating system you choose, different numbers of processors are available for use.

Operating system	Maximum number of processors
------------------	------------------------------

Operating system	Maximum number of processors
Windows NT 4.0, Enterprise Edition	8
Windows 2000 Advanced Server	8
Windows 2000 Datacenter Server	32
Windows 2003 Enterprise Edition	8-Way SMP
Windows 2003 Datacenter Edition	32-Way SMP

Operating systems Not Supported	Description
Windows 2003 with Terminal Services enabled	KB Article 327270: SQL Server 2000 Is Not Supported on Windows Server 2003 Terminal Server.

Testing an application's performance in a lab or some other controlled environment

Clustering from an operating-system perspective requires Windows NT 4.0 Server, Enterprise Edition, Windows 2000 Advanced Server, or Windows 2000 Datacenter Server. Windows 2000 Datacenter Server provides the most comprehensive solution; it is designed specifically for high availability. It requires a Service Level Agreement. If the operating system is not listed above, it does not support more than four processors.

Memory

Depending on the operating system that is used, SQL Server 2000 can take advantage of different amounts of maximum memory. An installation of SQL Server 2000 Enterprise Edition supports up to 32 gigabytes (GB) of memory on Windows 2000 Datacenter Server, without Address Windowing Extensions (AWE) enabled. The following table shows the maximum amount of memory available to SQL Server 2000 per operating system.

Operating system	Maximum
Windows NT 4.0, Enterprise Edition	3 GB
Windows 2000 Advanced Server	8 GB (with AWE enabled)
Windows 2000 Datacenter Server	64 GB (with AWE enabled)
Windows 2003 Enterprise Edition	64 GB (with AWE enabled)
Windows 2003 Datacenter Edition	64 GB (with AWE enabled)

Address Windowing Extensions and Physical Addressing Extension Memory

With AWE, a memory-intensive application can now run much more efficiently under SQL Server 2000 to increase performance. Windows 2000 Advanced Server and Windows 2000 Datacenter Server introduced the enhanced AWE API. AWE allows applications to access large amounts of physical memory. Due to limitations of 32-bit memory addressing, Windows NT 4.0 and Windows 2000 without AWE enabled can use only up to 4 GB of physical memory. By default, 2 GB of memory is dedicated to the operating system and 2 GB of memory to the application. With a **/3GB** switch in the Boot.ini used by the operating system, an application such as SQL Server can access up to 3 GB of memory, and the operating system is reduced to 1 GB of memory. As a result, even if a server were configured with 8 GB of memory, anything beyond 4 GB would have been virtually unusable. AWE is the support built into the operating system as a way of exposing extended memory to Win32®-based applications.

AWE requires an application, such as SQL Server 2000, to be coded specifically for AWE. AWE support within SQL Server 2000 must be configured using the **awe enabled** option in **sp_configure**. This is set per instance. By default, **awe enabled** is set to 0, or off. Enabling AWE support in SQL Server 2000 also requires some additional operating-system configuration. For more information, see "AWE Memory" in SQL Server Books Online.

Another option you can use to take advantage of larger amounts of memory is Physical Addressing Extension (PAE). PAE enables a 32-bit operating system to address memory above 4 GB. For information about PAE including how to set it up, see the Knowledge Base article "268363 – Intel Physical Addressing Extensions (PAE) in Windows 2000" at <http://support.microsoft.com/default.aspx?scid=kb;en-us;268363&sd=tech>

Note: If PAE is enabled, you may encounter backup and restore errors with Windows 2000 or SQL Server 2000 backups. See the Knowledge Base article "280793 – SQL Server 2000 or Windows 2000 Backup Not Viewable

While Running in PAE Mode" at <http://support.microsoft.com/default.aspx?scid=kb;en-us;280793&sd=tech>

When choosing hardware for your cluster solution, if you plan to use large memory, make sure that the configuration includes hardware that supports large memory. To check, search all categories for the term "large memory" on the HCL.

The following table summarizes how extended memory settings should be configured based on the amount of large memory you are setting up.

4 GB or less	4 GB to 8 GB	More than 8 GB
/3GB switch	/3GB enabled	/3GB disabled
	AWE enabled	AWE enabled
	PAE enabled (Boot.ini)	PAE enabled (Boot.ini)

Note: If you are enabling AWE or PAE memory, it is highly recommended that the configuration be tested prior to bringing the server(s) online in a production capacity.

The three memory options are enabled with two different mechanisms.

/3GB

The /3GB option is a switch enabled through the boot.ini file. After you have installed Windows 2000 Advanced Server, modify the boot.ini file to add the /3GB parameter to the ARC path, as shown in bold format in the following example:

```
mul ti (0)di sk(0)rdi sk(0)parti ti on(2)\WI NNT="Wi ndows 2000 Advanced Server"  
/3GB /basevi deo /sos
```

PAE

PAE is also enabled via a switch in the boot.ini. Open the boot.ini file, and add the /PAE parameter to the ARC path, as shown in bold format in the following example:

```
mul ti (0)di sk(0)rdi sk(0)parti ti on(2)\WI NNT="Wi ndows 2000 Advanced Server"  
/PAE /basevi deo /sos
```

AWE

AWE is enabled within a SQL Server 2000 query by a call to *sp_configure*, as shown in this example:

```
EXEC sp_confi gure ' awe enabl ed' , 1  
RECONFI GURE
```

When you implement AWE memory, consider these issues:

- The instance of SQL Server does not dynamically manage the size of the memory address space used.

When AWE is enabled with SQL Server 2000, if the **max server memory** configuration option is not set, SQL Server grabs the total memory available (except 128 MB to allow the base operating system to function), potentially depriving the operating system and any other processes that would be running on the same server.

- After it has been initialized, AWE memory holds all the physical memory acquired at startup until it is shut down.

If AWE is enabled and is taking too much memory, SQL Server must be shut down to reconfigure it, causing downtime (which makes a high availability option such as failover clustering less available). Because the memory pages used by the instance of SQL Server are taken from the nonpageable pool of Windows memory, none of the memory can be exchanged. This means that if the physical memory is filled up, SQL Server cannot use the page file set up on a physical disk to account for the surplus in memory usage.

- Once the **max server memory** option is configured, set the working set size to 0.

For more information about configuring AWE memory on your server, see "Using AWE Memory on Windows 2000" in SQL Server Books Online, as well as the following:

- PAE Server Design, which includes links to white papers (including "Supporting PAE Memory Under Windows 2000") at <http://www.microsoft.com/whdc/system/platform/server/pae/default.mspx>
- The "Address Windowing Extensions and Windows 2000 Datacenter Server" white paper at <http://www.microsoft.com/whdc/system/platform/server/pae/default.mspx>

Networking

Auto sensing of the network cards should be set to static speeds that match your LAN or WAN network. For example, set all network cards to 100 megabit with full duplex if that is how your network is configured. SQL Server 2000 supports multiple IP addresses (each on a different subnet) and network cards. If larger bandwidth is required, SQL Server 2000 has support for higher bandwidth networking with Giganet or Compaq's Servernet II technology on Compaq hardware. If these technologies are used, they will create higher performance between multiple SQL Servers. Giganet support is built-in, and the update to enable Servernet II is located at <http://www.microsoft.com/downloads/details.aspx?FamilyID=790c837f-7c55-4c86-b10c-31ada2accf43&displaylang=en>

Node Location

Due to certain limitations, such as the physical restrictions on distance supported by SCSI or FibreChannel, the nodes in a failover cluster must be located near each other. However, a server cluster is unaware of distance, so in theory, the nodes can be located anywhere. If a geographically dispersed cluster is to be configured, consider the following points:

- The private and public network connections between cluster nodes must appear as a single, nonrouted local area network (LAN) using technologies such as a virtual LAN (VLAN). In these cases, the network must guarantee a maximum round-trip latency between nodes of no more than 500 milliseconds for a connection. The cluster interconnect must appear as a standard Local Area Connection.
- Any geographically replicated storage technologies must preserve single disk semantics, such as persistent arbitration of a logical unit to Windows Clustering. The quorum disk must be replicated in real-time, synchronous mode across all sites.

Configuring a geographically separate cluster is complex and involves careful planning. Consult the hardware vendor for your cluster solution prior to implementing. It is also imperative that the hardware and software configuration is on the Hardware Compatibility List and purchased as a cluster solution in order to be supported by Microsoft. Supported geographically dispersed clusters can be found under the "Cluster/Geographic" category on the HCL. For more information on geographically dispersed clusters, see 280743 "Windows Clustering and Geographically Separate Sites" at: <http://support.microsoft.com/default.aspx?scid=kb:en-us:280743&sd=tech>

Another option that will provide high availability across different geographic locations is to employ log shipping, which is a feature of SQL Server 2000 Enterprise Edition. Log shipping is a process in which a transaction log from one server is applied to another server on a scheduled basis. Log shipping supports geographically separate locations, making it ideal for removing a single point of failure and protecting against data loss due to such events as a natural disaster. For more information about log shipping, see SQL Server 2000 Books Online or Chapter 13, "Log Shipping," in the *Microsoft SQL Server 2000 Resource Kit*.

Which option is the best for you to implement? Configuring a distance solution other than log shipping takes careful planning and tuning of your network, even if a third-party solution is involved. Although log shipping does require connectivity between the locations, it is not bound to the 500-millisecond limitation; therefore, there can be a higher latency, for example, if you are shipping logs from London to San Francisco. Both solutions, however, would need to have a proper plan in place to put into action in the event of a failure. Log shipping does require more manual intervention and administration than a clustered solution, because log shipping does not automatically perform the role change required to bring the warm standby online.

Configuration Best Practices

In addition to understanding the fundamentals of failover clustering, you may find it useful to keep the following tips and best practices in mind when configuring your servers.

Disk Configuration and File Placement

The main component of any database system is its storage — it contains the valuable data used and inserted by an application. For high availability, disks used by SQL Server for data and log must be part of a fault tolerant external array. The disks should be high speed for performance and support both large amounts of I/O and large amounts of storage space to allow your databases to grow over time. Keep in mind that in a failover cluster, the

shared cluster disk array is a single point of failure. One way to mitigate this risk is to stock spare hard drives in a closet in the event of a failure.

The disks can be configured either with a small computer system interface (SCSI) or FibreChannel. FibreChannel is the recommended method of implementing a shared disk array. FibreChannel is designed specifically for high bandwidth and high capacity. Storage Area Networks (SANs) are disk arrays that use networking protocols over FibreChannel to do all I/O. Use of SANs is supported for use in conjunction with failover clustering as a Cluster/Multiple-Cluster Device if purchased as part of a complete clustering solution.

Windows Clustering can be used in a SAN environment. The HCL category cluster/multi-cluster device lists the set of SAN-capable storage devices that are supported and have been tested as SAN storage units with multiple MSCS clusters attached. By cross-matching the devices on this list with the complete cluster configurations defined in the cluster HCL category it is possible to deploy a set of Windows servers and clusters on a SAN fabric with shared storage devices in a way that is supported by Microsoft. For more information on SAN support with clusters, please see the Knowledge Base article "304415 – Support for Multiple Clusters Attached to the Same SAN Device" at <http://support.microsoft.com/default.aspx?scid=kb;en-us;304415&sd=tech>

Note: SCSI is not supported in a Windows 2000 Datacenter Server cluster. FibreChannel must be used.

Note: Network Attached Storage (NAS) devices are not supported in a clustered environment. For more information, see the Knowledge Base article "304261 – Support for Network Database Files" at <http://support.microsoft.com/default.aspx?scid=kb;en-us;304261&sd=tech>

Data and log devices, as well as tempdb, should be placed on separate disks using as many different channels as possible, with the caveat that this will limit the number of instances that can be installed on the cluster. If your system is very large, or has hotspots, you may decide to use filegroups as a method of splitting up the disk I/O. Further segmenting this by putting filegroups on different disks, on different channels, can result in a performance boost. It is important to keep the file placement and channel usage in mind when you are analyzing your high-availability design. Performance issues caused by bottlenecks can be incorrectly perceived as availability problems. Physical file/disk layout problems may require downtime to resolve, which would lower the availability of your system.

Data drives should use the RAID configuration of striped mirrors for maximum availability. This means that first the drives are each mirrored and then the whole thing is striped. Striped mirrors are sometimes known as RAID 1+0. The next best for availability, which also gives a little better performance than striped mirrors, is mirrored stripes, which is sometimes known as RAID 0+1. Mirrored stripes mean that a group of disks is striped as one set, and then mirrored. It cannot tolerate failures as well as striped mirrors. RAID 5, which has been a popular option for years, does not provide the highest availability or performance. It is a good option if striped mirrors or mirrored stripes are not available, either because of money or hardware constraints.

Log drives can be configured either with RAID 1, which is plain mirroring, or striped mirrors/mirrored stripes. Logs are important to protect, so choose the appropriate availability of the disk for your high availability plan.

Note: The terminology for striped mirrors and mirrored stripes (i.e. 0+1, 1+0) may vary from manufacturer to manufacturer.

Important: Here are some caveats for disk usage in a server cluster:

- Dynamic disks are not supported in Windows Clustering. For more information, see the Knowledge Base article "237853 – Dynamic Disk Configuration Unavailable for Server Cluster Disks" at <http://support.microsoft.com/default.aspx?scid=kb;EN-US;237853>.
- File compression is not supported on a clustered database server.
- Software RAID is not supported in a cluster; hardware RAID must be used.
- On a clustered instance of SQL Server 2000, data storage on mounted drives is not supported. This applies to Windows 2000 and to Windows Server 2003. For more information, see the Knowledge Base article "819546 SQL Server 2000 support for mounted volumes" at <http://support.microsoft.com/?id=819546>.

For additional information and configuration worksheets, see the "Shared Cluster Disk Partition Configuration Worksheet" and the "SQL Server 2000 Failover Cluster Disk Configuration Worksheet" located in Appendix C.

Quorum Disk

Do not put any database files, such as data or log files, on the quorum disk. By default, SQL Server 2000 Setup does not use the quorum disk, unless there are no other disks available. From a physical disk standpoint, if it is possible, the quorum disk should be on a separate spindle and be on a separate drive from the SQL Server data.

Controller Configuration

Choose a card with enough channels to split the logical grouping of disks (for example, data and logs) to reduce I/O contention; however, this will limit the number of virtual servers you can install. If the FibreChannel/RAID controller is internal to the node and not in the shared disk array, writeback caching should be disabled. Because even with battery backup, once the resources fail over to another node, there might be items still in the cache. If the services are failed back over to the node, corruption may occur because the controller will attempt to overwrite things on the disk. Data loss would also occur in a failover if transactions were in the cache but not processed.

Using more than one RAID controller not only may increase performance and reduce I/O contention (which will increase availability), but the redundancy at the hardware level also gives you higher availability in the event one of the RAID controllers fails.

Making Sure Logical Disks Are Seen by the Virtual Server

In failover clustering, if the virtual server cannot "see" the disk resources, which are the logical drive letters, the virtual server will not operate properly. This is probably caused by one of two things:

- The proper disk drivers might not be installed.

Make sure that they are installed. In some cases (for example, an operating system upgrade from Windows NT 4.0, Enterprise Edition to Windows 2000 Advanced Server), there may be specific drivers for Windows 2000, but the old drivers may still be on the system.

- The drive might not be a dependency of the SQL Server 2000 virtual server. During the installation of the virtual server, only one data drive is can be selected, so if multiple drives are needed by the SQL Server 2000 virtual server, you will need to add the dependency following the successful completion of setup. To check whether the drive is a dependency of the virtual server, see the section "SQL Server 2000 Failover Cluster Dependencies" later in this paper.

Adding A Logical Disk To Your Cluster Configuration

Because dynamic disks are not supported in Windows Clustering, adding a disk to the configuration at some point after the initial configuration will incur some downtime. Consult the Knowledge Base article "175278 – How to Install Additional Drives on the Shared SCSI Bus" for instructions on adding drives to your cluster configuration. It can be found at <http://support.microsoft.com/default.aspx?scid=kb:en-us:175278&sd=tech>

Once the drive is recognized at the Windows Clustering level, take the SQL Server cluster resource offline, and add the drive as a dependency in Cluster Administrator. After bringing the SQL Server resource online, the SQL Server virtual server will now be able to use this new drive.

Expanding An Existing Logical Disk In Your Cluster Configuration

It is possible to expand the existing disk space at the hardware level on a defined cluster disk. Consult the Knowledge Base article "263590 – How to Extend the Disk Space of an Existing Shared Disk with Windows Clustering" for instructions on expanding drives in your cluster configuration. It can be found at <http://support.microsoft.com/default.aspx?scid=kb:en-us:263590&sd=tech>

Keep in mind that this will incur some downtime, and it must be planned as to not affect the availability of your end users.

Cluster Nodes and Windows Domains

All nodes in the cluster must be members of the same domain and able to access a domain controller and a Domain Name System (DNS) server, as well as a WINS server. The nodes should not be configured as domain controllers if you are going to install SQL Server, as you may encounter some problems, including the fact that domain controller functionality (such as Active Directory) is not cluster aware, so all information will be local. This impacts things such as a directory enabled program's ability to publish under the virtual server's computer object, which does not work in a cluster. A WINS server is still required for Windows 2000/SQL Server 2000 in a clustered environment if name resolution of virtual resource is required.

Using Multiple IP Addresses

When configuring a network card for use within a cluster, you should consider the options available based on how many types of networks you must support, given the number of network cards available. Also, keep in mind assigning more IP addresses you assign to allow connectivity to SQL Server may affect the availability of your failover cluster, as in some cases, you may not have control of the routers to re-route networks.

For example, you may attempt to maximize the use of your network cards by configuring one card for all

communications including:

- All external client connections and inter-node traffic.
- Internal cluster (private network between the cluster nodes only).
- Client access only (public network to allow client connectivity).

Even though only one network card in a cluster can usually handle all cluster network communications, this creates a single point of failure. The optimum configuration is to have a separate network card to handle each type of connectivity.

Ideally, three IP addresses on separate subnets and three network cards should be associated with any instance of SQL Server:

- **Heartbeat**

This should be configured as "internal cluster communications only" to allow the nodes to communicate amongst themselves with no additional traffic from external clients. This should also be on an opposite IP class. For more information, see the Knowledge Base article "258750 – Recommended Private 'Heartbeat' Configuration on a Cluster Server" at <http://support.microsoft.com/default.aspx?scid=kb;EN-US;258750>

- **Client Connectivity**

If you have only one IP address that will be configured for client access, configure it as "all communications", as it will provide redundancy for the internal communications in the event that the Heartbeat fails. If there is more than one IP address that will be used by clients, you can configure them either as "all communications" or "client access only."

- **Separate Private Network**

This is different from the heartbeat and should be configured as "client access only." This should be configured so only the servers in the cluster will be able to access this particular IP address. This configuration will enable files to be transferred, or optimally, allow log shipping to be configured in a way that will not affect the heartbeat or client network traffic.

Not all three IP addresses can be on the same subnet. Connectivity problems may be encountered if the same subnet is used for more than one IP address, even if it is not currently being used in the cluster. For example, the following table shows both a correct and an incorrect server configuration.

Network card	Correct configuration	Incorrect configuration
1 – Configured as Public Network	172.21.10.1	172.22.10.1
2 – Configured as Public Network	172.22.10.2	172.22.10.2
3 – Configured as Private Network	172.23.7.3	172.23.7.3
4 – Configured as Heartbeat	10.10.10.1	172.24.2.5

In addition, there are network cards that support multiple IP addresses being bound to them. Although this allows a failover cluster to talk over more than one network, it is potentially a single point of failure, which must be avoided in a high availability solution. Therefore, always ensure you have at least one network card for each required function, even if the card can support multiple IP addresses.

For more information, see the Knowledge Base article "175767 – Expected Behavior of Multiple Adapters on Same Network" at <http://support.microsoft.com/default.aspx?scid=kb;en-us;175767&sd=tech>

Memory Configuration

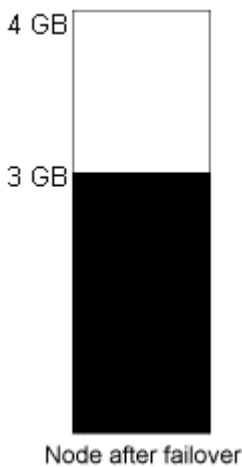
This section presents considerations for memory usage in a SQL Server 2000 failover cluster.

Single-Instance Failover Cluster

In a single-instance SQL Server 2000 failover cluster, the failover scenario is simple: If the primary node fails, all processes go to the designated secondary node configured (see "Configuring Node Failover Preferences" later in this paper). The secondary node should always be configured exactly the same as Node A in terms of hardware. If not, problems may occur if the failover node does not have the same capacity as the primary node, especially in terms of memory, as evidenced in example two. Consider any other processes that may be running on the server node, as well as overhead for the operating system.

Example One: Two Nodes, Exact Configuration

Think of your cluster nodes as two glasses of water. The glasses can each hold 4 ounces of water. Glass A has 3 ounces of water, and Glass B has no water. If you pour the water from A into B, it will all fit with no problems. In the case of a SQL Server 2000 failover cluster, the resources will function as they did on the primary node. The following illustration shows this scenario.



Example Two: Two Nodes, Unequal Configuration

Again, think of your two cluster nodes as glasses of water. Glass A has a capacity of 4 ounces. It is filled with 3 ounces of water. Glass B has a capacity of 2 ounces. If you pour the water from A into B, it will overflow and spill because it cannot hold all of the liquid that was in Glass A. Therefore, if your failover node does not have the physical memory to support the instance of SQL Server, paging to disk will occur because SQL Server is looking for more memory than is physically available. The server will now be short of resources, potentially causing the node to become unresponsive. Figure 3 illustrates this scenario.

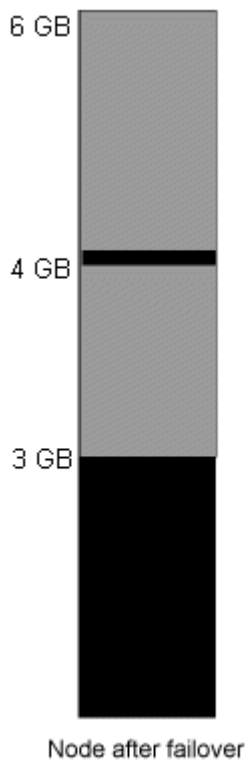


Multiple-Instance Failover Cluster

In a multiple-instance SQL Server 2000 failover cluster, the scenario becomes more complex. With up to 16 instances that can be active at a time on one node, how does one effectively manage memory? First and foremost, ensure that all servers have the same amount of memory, and that it is enough to handle the instances that could potentially fail to that node. Another important consideration is to cap memory usage of the instance of SQL Server 2000 with **max server memory** (see "Address Windowing Extensions and Physical Addressing Extension Memory" earlier in this paper). Especially if AWE memory is enabled, **max server memory** must be set in a multiple-instance cluster to prevent starving the server node, as shown in example two that follows. Consider any other processes that may be running on the server, as well as overhead for the operating system.

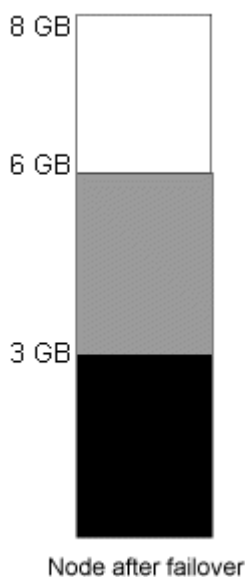
Example One: Two Instances of SQL Server, Noncapped Memory

Once again, consider the two glasses of water. Both glasses have a maximum capacity of 4 ounces. Glass A and Glass B contain 3 ounces of water each. If you pour the contents of Glass B into Glass A, only 1 ounce will fit before an overflow of the remaining 2 ounces occurs. Similar to the previous example, if the failover node does not have the physical memory to support the second instance of SQL Server 2000, paging to disk will occur because SQL Server 2000 is looking for more memory than is physically available. The server will now be short of resources, potentially causing the node to become unresponsive. The following illustration shows this scenario.



Example Two: Two Instances of SQL Server 2000, Capped Memory

Again, think of your two cluster nodes as glasses of water. Both glasses have a maximum capacity of 8 ounces. Glass A and Glass B contain 3 ounces of water each. If you pour the contents of Glass B into Glass A, Glass A can handle the entire amount of liquid with no overflow. From a SQL Server perspective, for this example to work, AWE memory must be enabled, and each instance must use the **sp_configure** stored procedure **max server memory** option to cap memory on each instance at 3 GB. In the event of the failover, there is still 2 GB of memory left for the operating system and any other processes running. The following illustration shows this scenario.



Processor Capacity

Although there are no specific requirements as to how much processor power you will need for SQL Server 2000, since it is dependent upon how your application utilizes SQL Server, each cluster node should be configured with enough processors of sufficient power to handle the load for any instance that may run on the node. Unless processor affinity is set for the virtual server, all instances will share the processors in the server. The best way to determine how much processing power is needed is to test your application with load prior to being rolled out in production, and to monitor it using System Monitor.

For example, you have one application that utilizes a virtual server. It is an OLTP application that constantly

utilizes all four processors of one server at a rate of approximately 75 percent. If a second virtual server in your failover cluster exhibits similar numbers and is set to failover to the same node as the first virtual server, the server may become slow, or potentially unresponsive, because it cannot handle the workload of the two systems. Instead of being memory starved, you will be CPU starved.

Using More Than Two Nodes

When you use more than two nodes on a SQL Server 2000 failover cluster, consider the following questions:

- How much memory should be configured for each instance?
- What nodes are the failover cluster nodes for the particular instance? What is the preferred order?
- Is there enough disk space and memory to support every instance configured to fail over to a particular node?
- Is the hardware configured to support failover clustering without affecting other instances?

Because SQL Server 2000 can use four nodes when supported by the operating system (the number of virtual servers is only limited by the choice of operating system and the capacity of your hardware), and have up to 16 instances, these considerations become more important as mission-critical systems become larger. Although SQL Server can support up to the 16-instance limit, having more than four (which is a 1:1 ratio for virtual servers to nodes in a Windows 2000 Datacenter Server cluster) is not recommended. Another consideration is the number of logical drives that can be assigned — because each instance would require its own dedicated drive letters. There is a limit to the number of drive letters available because of the finite size of the English alphabet. If multiple drive letters were assigned to each individual instance, it would greatly reduce the number of instances that can be created.

As noted earlier in this paper, it may be necessary to assign a designated unique port to a SQL Server 2000 virtual server after installation. By default, SQL Server 2000 will dynamically assign a port during installation of the virtual server. To change the port manually, use the Server Network utility.

Scenario One: Four-Node Multiple-Instance SQL Server 2000 Failover Cluster, Three Active Nodes, One Standby (N+1)

With four-node support, Windows 2000 Datacenter Server provides more flexibility in terms of a cluster configuration. The recommended way of using a four-node Windows 2000 Datacenter Server cluster in a SQL Server environment is to have three of the nodes each owning an instance of SQL Server 2000 and have the fourth be the warm standby. This is not unlike a log shipping scenario, or a single-instance failover cluster in which at least one node is waiting for work. This scenario is known as N+1. Instead of configuring your failover cluster to allow the instances to fail first to a node with another instance of SQL Server 2000 running, the fourth node should be configured as the primary failover. This would reduce the issue of having too many instances starving the resources of one node. AWE memory should be enabled in this scenario to allow each instance of SQL Server to address more memory than the 1 GB currently available. This allows your applications to scale out rather than limiting them if they exceed the memory allocation for SQL Server.

Scenario Two: Four Node Multiple-Instance SQL Server 2000 Failover Cluster, All Four Nodes Active

Running four instances of SQL Server 2000 on four nodes requires careful planning, so that another instance will not starve resources due to memory and processor consumption in the event of a failover. Memory is not as much of an issue as processor resources. For example, if the workload on the production online transaction processing (OLTP) system regularly uses eight processors at 50-percent utilization and all four active instances of SQL Server 2000 demonstrate similar behavior, memory can only compensate for processor so much; more processors must be added.

Miscellaneous Configuration Issues

- Disable or do not install antivirus software on your cluster. For more information, see 250355 "Antivirus Software May Cause Problems with Cluster Services" at: <http://support.microsoft.com/default.aspx?scid=kb;en-us;250355&sd=tech>
- It is not recommended to have both SQL Server 2000 and Microsoft Exchange 2000 on the same cluster.
- Make sure all instances of SQL Server have their own unique Network Name and IP address.
- When configuring replication with clustered servers, create a MSCS file share for replication to be configured against so all cluster nodes will be able to access it in the event of a failover.
- It is not recommended that any file shares be used on the same cluster disks that a SQL Server is using.
- WINS is required for NetBIOS name resolution of all virtual resources.

- Resolve any potential application issues such as locking and blocking that may cause availability problems.

For additional considerations, see SQL Server 2000 Books Online.

Sample SQL Server 2000 Failover Cluster Configurations

There are, of course, many different ways to configure your failover cluster depending on your system requirements and the hardware you have available. In situations where you have detailed information on your systems average and peak throughput, always do proper capacity planning for your servers. For detailed coverage of capacity planning techniques, see the *Microsoft SQL Server 2000 Administrator's Companion*.

OLTP System Server Layout

This design is for a classic OLTP application. The transaction log for is split over a set of disks to support a high volume of transactions per second. Both servers are configured exactly the same:

- Operating system: Windows 2000 Advanced Server
- Number of nodes: Two
- Number of processors (per server): Eight
- Memory (per server): 4 GB
- SQL Server memory configuration: limited to 3 GB
- Internal disk configuration for the operating system: Two to four internal drives (9 GB each) with RAID 1. For four or more drives, use RAID 0+1.

Shared FibreChannel SAN Configuration

	GB (total capacity)	Total disks (external; 18 GB each; RAID 0+1)	Files on drive
Drive Q	36	4	Quorum Drive
Drive R	54	6	Transaction Log
Drive S	216	24	SQL Server Data Files, tempdb
Drive T	36	4	Backups/Imported Data Files (could use larger disks)

Multiple-Instance Failover Cluster with Log-Shipped Standby Server

A common high availability scenario is to use failover clustering as the primary method, but also sending the transaction logs to a completely different server as another disaster recovery method. This server (known as a warm standby) should be located in another geographic data center away from the failover cluster to avoid a single point of failure. However, there must be good network connectivity between the locations. Log shipping is a feature of SQL Server 2000 Enterprise Edition. For more information about log shipping, see Chapter 13, "Log Shipping," in the *Microsoft SQL Server 2000 Resource Kit* and SQL Server Books Online.

Multiple-Instance Failover Cluster

To support future growth, an eight-way box is selected, but only four processors are added. Both instances support OLTP applications. Instance 1 is replicated to a report server (not shown here). Instance 2 is extracted for data warehousing on a weekly basis and is not replicated. Because of this difference, another mirrored set is added to Instance 1 transaction log.

- Operating system: Windows 2000 Advanced Server
- Number of nodes: Two
- Number of processors (per server): Four
- Number of SQL Server 2000 instances: Two
- Memory (per server): 4 GB, SQL Server limited to 1.5 GB per instance
- Internal disk configuration for the operating system: Two to four internal drives (9 GB each) with RAID 1. For four or more drives, use RAID 0+1.

Shared FibreChannel SAN Configuration

	GB (total capacity)	Total disks (external; 18 GB each)	Files on drive
Drive Q	36	4	Quorum Drive
Drive R	54	6	Instance 1: Transaction Log
Drive S	36	4	Instance 2: Transaction Log
Drive T	72	8	Instance 1: Data Files
Drive U	72	8	Instance 2: Data Files
Drive V	36	4	Instance 1: tempdb
Drive W	36	4	Instance 2: tempdb
Drive X	54	6	Backups/Imported Data Files (could use larger disks here)

Warm Standby Server Configuration

The standby server must have enough memory and processing power to support the workload of both databases in the event of a failover.

- Operating system: Windows 2000 Advanced Server
- Number of processors: Four
- Memory: 4 GB, SQL Server has 3 GB allocated to it.

Disk Configuration (RAID 1)

	GB	RAID partition	Total disks	Files
Drive C	18	A	2 internal	Operating system, page file, SQL Server executables, and system databases
Drive Z	54	C	6 internal	Backups/Imported data files
Drive T	36	E	4 internal	Transaction Log
Drive I	180	D	12 external	Data files, tempdb

In this scenario, there are less drives available on this computer. The data fits easily on the standby system, and the throughput requirements do not tax the production drive capacity. In reality, however, you should test to ensure that the warm standby can handle the workload. Not only should a disaster recovery plan be in place, but also plan to update the standby server in the event the requirements change.

Multiple-Instance Windows 2000 Datacenter Cluster (N+1 Scenario)

In this scenario, there are four servers of similar internal disk configuration, which share an external FibreChannel SAN. Three instances of SQL Server 2000 are active in the failover cluster. The requirements for CPU and RAM will vary depending on what role the server plays in the cluster. Three of the failover cluster nodes are the same, and own one instance each. The fourth node is the designated failover node and will have a larger capacity in the event all three instances fail. AWE memory is used. A failover cluster requires a well thought-out and certified hardware solution. For more information about failover clusters, AWE memory, and the N+1 configuration, see the section "Using More Than Two Nodes" earlier in this paper.

All Instances

- Operating system: Windows 2000 Datacenter Server

Active Instances

- Number of nodes: Three
- Number of processors (per server): Eight

- Memory (per server): 6 GB, SQL Server limited to 4 GB
- Internal disk configuration for the operating system: Two to four internal drives (9 GB each) with RAID 1. For four or more drives, use RAID 0+1.

Failover Node

The failover node must have enough memory and CPU to support all three active instances in case of failover.

- Number of processors: 32
- Memory: 16 GB
- Internal disk configuration for the operating system: Two to four internal drives (9 GB each) with RAID 1.

Shared FibreChannel SAN Configuration

	GB (total capacity)	Total disks (external; 18 GB each; RAID 0+1)	Files on drive
Drive Q	36	4	Quorum Drive
Drive T	36	4	Instance 1: Transaction Log
Drive U	36	4	Instance 2: Transaction Log
Drive V	36	4	Instance 3: Transaction Log
Drive I	90	10	Instance 1: Data files Instance 1: tempdb
Drive J	108	12	Instance 2: Data Files Instance 2: tempdb
Drive K	162	18	Instance 3: Data Files
Drive L	72	8	Instance 1: Data Files, possibly indexes
Drive M	72	8	Instance 2: Data Files, possibly indexes
Drive N	72	8	Instance 3: tempdb
Drive Z	36	4	Backups/Imported Data Files

For this example, Instance 1 and 2 are OLTP applications of similar access patterns. Instance 3 is an example of a decision support system (DSS) that uses tempdb heavily enough that you would want to move it to a different drive containing multiple fast disks. Note that, correspondingly, Instance 3 does not need more than the standard two disks for the transaction log drive. For more information, see *Inside SQL Server 2000 by Kalen Delaney from Microsoft Press®*, and Chapter 33, "The Data Tier: An Approach to Database Optimization," in the *Microsoft SQL Server 2000 Resource Kit*.

Because reporting systems use server resources differently from OLTP systems, it is important to take into account the characteristics of each workload. In the event of a failover where one failover cluster node may own both SQL Server 2000 virtual servers, can the system handle both from a memory, processor, and disk I/O standpoint? One node may be able to handle both for a short time. However, a disaster recovery scenario may require that the instance that was failed over to be failed back to the original node as soon as possible. Another option would be to allocate sufficient CPU and memory resources for each system and then limit resource usage to each instance.

[↑ Top of page](#)

Implementing SQL Server 2000 Failover Clustering

This section describes the implementation considerations when you configure a failover cluster. For installation instructions to install a new Windows 2000 server cluster, see <http://www.microsoft.com/technet/prodtechnol/windows2000serv/howto/clustep.mspx>

Restarting the server after installing SQL Server 2000 is recommended. This allows locked resources to be released and any pending file renames to be completed.

For information on setting up a one-node cluster for development purposes only using Windows NT 4.0, Enterprise

Edition or Windows 2000 Advanced Server, see the Knowledge Base article "245626 – INFO: Use the '-localquorum' Switch to Install a Single-Node MSCS Cluster" at <http://support.microsoft.com/default.aspx?scid=kb;en-us;245626&sd=tech>

Prerequisites

Prior to installing SQL Server 2000, make sure there are no errors in Event Viewer that may prevent a successful cluster installation. Verify that only the services necessary for the operating system are running. Any other services should be stopped because they may interfere with the installation process. These services include SNMP, the World Wide Web Publishing service, and vendor specific programs. The easiest way to start and stop multiple services is to create two batch files: one that contains multiple **net stop** commands and one that contains the corresponding **net start** commands.

The following tables list the services that should be left running.

Windows NT 4.0 Server, Enterprise Edition

Alerter	Plug And Play
Cluster Service	Remote Procedure Call (RPC) Locator
Computer Browser	Remote Procedure Call (RPC) Service
Event Log	Server
License Logging Service	Spooler
Messenger	TCP/IP NetBIOS Helper
Net Logon	Time Service
Windows NT LM Security Support Provider	Workstation

Windows 2000 Advanced Server and Windows 2000 Datacenter Server

Alerter	Plug and Play
Cluster Service	Process Control
Computer Browser	Remote Procedure Call (RPC) Locator
Distributed File System	Remote Procedure Call (RPC) Service
Distributed Link Tracking Client	Remote Registry Service
Distributed Link Tracking Server	Removable Storage
DNS Client	Security Accounts Manager
Event Log	Server
License Logging Service	Spooler
Logical Disk Manager	TCP/IP NetBIOS Helper
Messenger	Windows Management Instrumentation
Net Logon	Driver Extensions
Windows NT LM Security Support Provider	Windows Time Service
Network Connectors	Workstation

Installation Order

This section provides the installation order for a specific operating system and SQL Server 2000.

Windows NT 4.0 Server, Enterprise Edition

- Install Windows NT 4.0 Server, Enterprise Edition (do not install Microsoft Internet Information Server).
- Create domain users.
- Install Windows NT 4.0 Service Pack 3.
- Install Microsoft Internet Explorer 5.
- Disable NetBIOS on internal private networks.
- Install MSCS on both nodes.
- Manually create MS DTC Cluster Resources, see "Creating the MS DTC Resources (Windows NT 4.0, Enterprise Edition Only)" later in this paper.
- Install Windows NT 4.0 Option Pack if you want, but do not install MSMQ.
- Install Windows NT Service Pack 5 or later.

- Stop unnecessary services.
- Install SQL Server 2000 (see Appendix B – "Step-By-Step Installation Instructions for a New Virtual Server" for instructions).

Windows 2000 Advanced Server and Windows 2000 Datacenter Server

- Install Windows 2000 Advanced Server (Windows 2000 Datacenter Server is installed by the vendor if this is your choice of operating system).
- Install Microsoft Internet Explorer 5 Update (if necessary).
- Create domain users.
- Disable NetBIOS on internal private networks.
- Install Windows Clustering on one node.
- Join the other node(s) to the cluster.
- Run comclust.exe on all nodes to create the clustered MS DTC resource. (for more information, see "Failover Clustering Dependencies" in SQL Server 2000 Books Online).
- Stop unnecessary services.
- Install SQL Server 2000 (see Appendix B – "Step-By-Step Installation Instructions for a New Virtual Server" for instructions).

Creating the MS DTC Resources (Windows NT 4.0, Enterprise Edition Only)

This section provides the instructions for configuring the MS DTC resources for servers running Windows NT 4.0, Enterprise Edition, which requires a more complex setup procedure than that for Windows 2000 Advanced Server or Windows 2000 Datacenter Server.

Configuring the MS DTC IP Address

1. In Cluster Administrator, select the disk group that contains the quorum disk resource. Right-click the disk group, and then rename it.
2. Select the disk group you want. On the File menu, click New, and then click Resource. In the New Resource dialog box, in the Name box, enter MSDTC IP Address; in the Resource Type box, select IP Address; and in the Group box, select the group you want. Click **Next**.
3. Both nodes of the cluster should appear as possible owners. If not, add the node(s), and click **Next**.
4. In the Dependencies dialog box, select the disk resource in the group you selected from the Available Resources box, and then click Add. The disk resource appears in the Resource Dependencies box. Click **Next**.
5. In the **TCP/IP Address Parameters** dialog box, enter the TCP/IP information. In the **Address** box, enter the static IP address (for example, 10.1.14.131); in the **Subnet** mask box, enter the IP subnet (for example, 255.255.255.0); in the **Network to use** box, select the cluster network you want. Click **Finish**.
6. A message appears confirming that the IP address is successfully configured.
7. In the Cluster Administrator window, the newly created resource appears in the right pane. To start the resource (which is currently offline), right-click the resource, and click **Bring Online**.

Configuring the MS DTC Network Name

1. In Cluster Administrator, on the **File** menu, point to **New**, and then click **Resource**.
2. In the **New Resource** dialog box, in the **Name** box, enter **MSDTC Network Name**; in the **Resource Type** box, select **Network Name**; and in the **Group** box, select the group you want. Click **Next**.
3. In the **Possible Owner** dialog box, both nodes of the cluster should appear as possible owners. If not, add the node(s), and click **Next**.
4. In the **Dependencies** dialog box, the MS DTC IP address resource you configured previously appears in the **Available** resources box. Select the resource, and then click **Add**. The resource appears in the **Resource** dependencies box. Click **Next**.

5. In the **Network Name Parameters** dialog box, enter **MSDTC**, and then click **Finish**.
6. A message appears confirming that the IP address is successfully configured.
7. In the Cluster Administrator window, the newly created resource appears in the right pane. To start the resource (which is currently offline), right-click the resource, and then click **Bring Online**.

Implementation Best Practices

This section highlights some best practices when implementing a SQL Server 2000 failover cluster.

Configuring Node Failover Preferences

When you use more than two nodes in a failover cluster, it is important to consider in the event of a failover, which node should own the SQL Server processes? With up to four nodes available, there should be an order that makes logical sense for the production environment. The failover preferences should be set for the group containing all the resources for the instance of SQL Server (not only on the virtual server) to ensure that all resources properly fail over to the same node. For example, in an N+1 configuration, each group would have the idle node second in the list of preferred owners. This means that if any of the nodes failed, the resources on that node would move to the idle node.

Important: Do not use Cluster Administrator to remove nodes from the resource definition. Use SQL Server Setup for that functionality, for instructions, see "Adding or Removing a Cluster Node from the Virtual Server Definition" later in this paper.

To configure the preferred failover order for the nodes

1. Start Cluster Administrator. Right-click the group containing the SQL Server 2000 virtual server, and then click **Properties**.
2. On the **General** tab, the **Preferred** owners list box displays all cluster nodes that can potentially own the processes in that group, and the current order in which they would fail over. To change the order, click **Modify**.
3. In the **Modify Preferred Owners** dialog box, make any changes to the preferred failover order. All nodes currently configured as potential owners will appear in the right pane in the order of failover preference. For example, there are four nodes in a cluster: **DENNIS**, **TOMMY**, **JAMES**, and **CHUCK**. All four nodes of the cluster can be potential owners, and the order of failover if **DENNIS** goes down is set to be **JAMES** then **TOMMY**, and finally **CHUCK** if both **JAMES** or **TOMMY** are unavailable.

Failover/Failback Strategies

An overall cluster failover/failback policy is recommended. Failovers can be controlled in terms of a threshold, meaning that after a certain point, a resource will not be failed over. There are two levels of thresholds: resource and cluster. Depending on how the resource is configured, it can affect the group failing over to another node.

In the event of a failover, the cluster group containing the SQL Server resources can be configured to fail back to the primary node when and if it becomes available again. By default, this option is set to off because usually there is no problem with continuing on the secondary node. This setting provides an opportunity to analyze and repair the problem on the failed node.

To configure automatic failback for a cluster group

1. Start Cluster Administrator. Right-click the group containing the SQL Server 2000 virtual server, and then click **Properties**.
2. In the **Properties** dialog box, click the **Failback** tab.
3. To prevent an automatic failback, select **Prevent Failback**. To allow automatic failback, select **Allow Failback**, and then one of the following options:

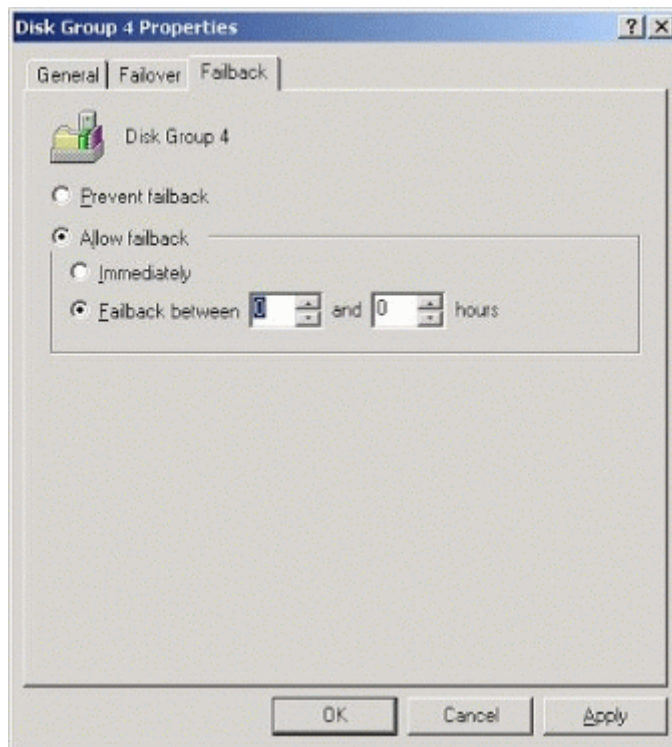
- **Immediately**

This means that the second Windows Clustering detects that the preferred cluster node is online, and it will fail back any resources. This is not advisable because it could disrupt clients and applications, especially at peak times in a business day.

- **Failback between *n* and *n1* hours**

This option allows a controlled failback to a preferred node (if it is online) during a certain period. The

hours are set using numbers from 0 through 23.



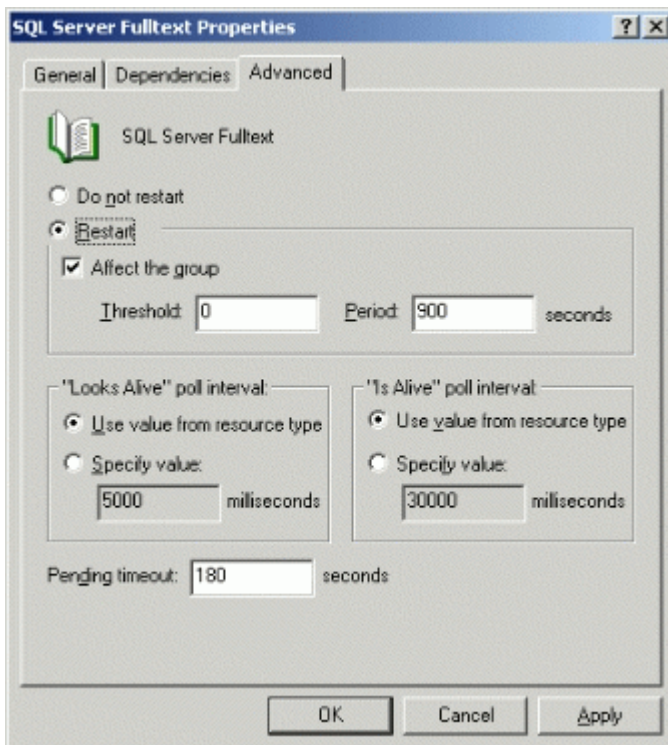
[See full-sized image.](#)

To configure thresholds for a resource

1. Start Cluster Administrator. Select the proper group containing the SQL Server 2000 virtual server, then right-click the resource to alter, and click **Properties**.
2. In the **Properties** dialog box, click the **Advanced** tab.
3. Select **Do not restart** if the Cluster service should not attempt to restart or allow the resource to fail. By default, **Restart** is checked.
4. If **Restart** is selected, configure the restart policy:
 - **Affect the group**

To prevent the failure of the selected resource from causing the SQL Server group to fail over after the specified number of retries (**Threshold**) has occurred, uncheck the **Affect the group** checkbox.
 - **Threshold** is the number of times the Cluster service will try to restart the resource, and **Period** is the amount of time (in seconds) between retries. Set the numbers accordingly for your availability requirements.

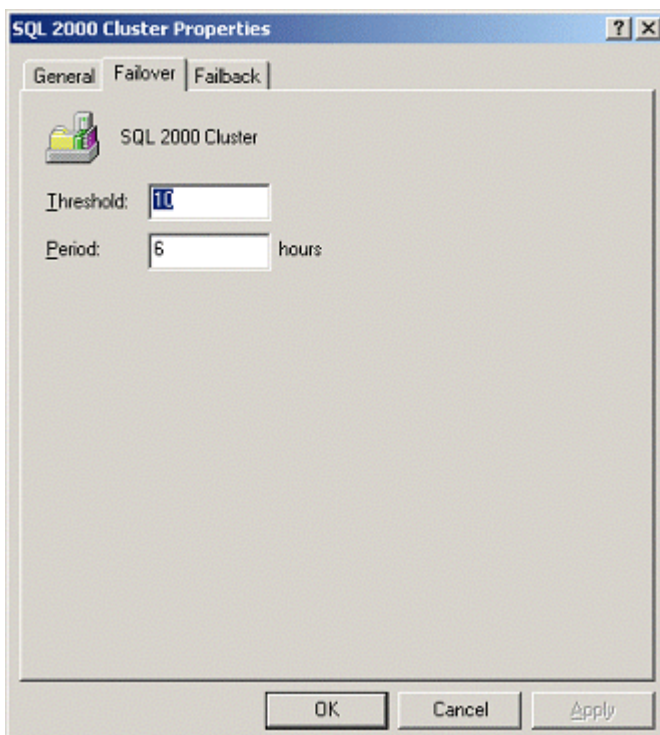
For example, if **Threshold** is set to 0, and **Affect the group** is selected, on detection of a failure, the entire group with the resource will be failed over to another node.
5. Do not modify the "Looks Alive" and "Is Alive" settings.
6. Unless necessary, do not modify **Pending timeout**. The value, represented in seconds, is the amount of time the resource in either the Offline Pending or Online Pending states has to resolve its status before the Cluster service puts it in either Offline or Failed status.
7. Click **Apply** and then **OK**.



[See full-sized image.](#)

8. To configure failover thresholds for a group. Start Cluster Administrator. Right-click the group containing the SQL Server 2000 virtual server, and then click **Properties**.
9. In the Properties dialog box, click the **Failover** tab.
10. To configure the failover policy, in the Threshold box, enter the number of times the group is allowed to failover within a set span of hours. In the Period box, enter the set span of hours.

For example, if **Threshold** is set to 10 and **Period** is set to 6, the Cluster service will fail the group over at a maximum of 10 times in a six-hour period. At the eleventh failover in that six-hour period, Windows Clustering will leave the group offline. This affects only resources that were failed over; therefore, if the SQL Server resource failed 11 times, it would be left offline, but the IP could be left online.



SQL Server 2000 Failover Clustering and MS DTC

There are two schools of thought when it comes to designing a server cluster when MS DTC is used:

- Use the default configuration, which will configure MS DTC to use the quorum drive. This is the most popular, and most often, recommended solution.
- Plan in advance, and create a separate cluster disk that is dedicated to MS DTC. While this may reduce contention on the quorum drive if MS DTC is being used in the cluster, it may mean that not enough drives will be available for the SQL Server instances. It also means a few more steps in configuring the cluster. For example, a clustered BizTalk Server configuration requires that MS DTC be placed on a separate drive and in a separate cluster group. For more information on BizTalk Server clustering considerations, see the link for the BizTalk Server clustering whitepaper in Appendix A.

Either way, make sure that the drive that is going to be associated with MS DTC is about 500 MB. The drive does not need to be gigabytes in size. If MS DTC is going to be placed on the quorum, make sure there is enough room for the cluster files as well. Even if MS DTC is not going to be used currently, it is easier to configure it during the initial installation in the event that it may be used at some point.

If for some reason the server cluster design requires another cluster disk for MS DTC, consult the Knowledge Base article "294209 – INF: Rebuilding or Moving MSDTC Used with a Failover Clustered SQL Server" at:

<http://support.microsoft.com/default.aspx?scid=kb;en-us;294209&sd=tech>

Other information about MS DTC and Windows clustering can be found in the Knowledge Base article "Microsoft Distributed Transaction Coordinator Recovery Techniques in Windows 2000 Cluster Server", which can be found at: <http://support.microsoft.com/default.aspx?scid=kb;en-us;243204&sd=tech>

Cluster Group Configuration for SQL Server 2000

When setting groups using Cluster Administrator, there is no minimum or maximum number required. From a conceptual standpoint, it makes sense to put similar items in one group, for example:

- Group the Cluster IP Address, cluster name, and quorum disk in one group.
- MS DTC by default should stay in its default location unless it needs to be moved.
- Group the SQL Server IP Address, SQL Server Network Name, SQL Server, SQL Server Agent, and SQL Server full-text resources for a specific instance in a group for each instance.

MS DTC may be placed in either the group with the main cluster IP, the quorum disk, or the group with the disk that MS DTC is configured to use. MS DTC is a shared resource and it depends on the disk it is configured to use. However, it is recommended that SQL Server or any other cluster applications not be put in the same group with the quorum disk. MS DTC would be the exception to this since this is where the operating system places it by default.

SQL Server 2000 Failover Cluster Dependencies

For the latest information on dependencies refer to Knowledge Base article, "835185 Failover cluster resource dependencies in SQL Server", <http://support.microsoft.com/?id=835185>.

Cluster resources may be dependent on other resources to start before they are brought online. A default installation has the following dependencies (this table is not meant to represent a group; only the default dependencies for each listed resource).

Resource	Dependencies
Cluster IP Address	None
Cluster Name	Cluster IP Address
Quorum	None
MS DTC	Cluster Name, Disk Resource (Quorum is the default)
SQL Server IP Address	None
SQL Server Network Name	SQL Server IP Address
SQL Server (the virtual server itself)	SQL Server Network Name, Disk Resources associated with the instance
SQL Server Agent	SQL Server

Resource	Dependencies
SQL Server FullText	SQL Server

One can add other dependencies to these resources, but for a SQL Server 2000 failover cluster to function properly, this is the base configuration that must not be altered. Remember that any customizations beyond the base setup may result in unplanned failovers if cluster dependencies are not properly configured.

To check dependencies of a resource

1. Start Cluster Administrator, right-click the resource, and then click **Properties**.
2. On the **Properties** dialog box, click the **Dependencies** tab, and then click **Modify**.
3. On the Modify **Dependencies** dialog box, the available resources for the cluster appear in the **Available** resources list. Select the drive to add, click the arrow to move the resource to the **Dependencies** list, and then click **OK**.
4. To verify that the resource is now a dependency, on the **Properties** dialog box, click the **Dependencies** tab.

SQL Server 2000 Analysis Services and Failover Clustering

The SQL Server 2000 Analysis Services (OLAP and data mining) component is not supported on a failover cluster. For more information, see the Knowledge Base article "254321 – Clustered SQL Server Do's, Don'ts and Basic Warnings" at <http://support.microsoft.com/default.aspx?scid=kb;en-us;254321&sd=tech>

For information on how to make SQL Server 2000 Analysis Services highly available, see the whitepaper <http://www.microsoft.com/downloads/details.aspx?FamilyID=9989a445-142b-4872-ac68-2b50f05228e2&displaylang=en>

SQL Mail and Failover Clustering

Using SQL Mail with a SQL Server 2000 virtual server is not completely supported because the underlying MAPI protocol that is used is not cluster-aware. For more information, see the Knowledge Base article "263556 – How to Configure SQL Mail" at <http://support.microsoft.com/default.aspx?scid=kb;en-us;263556&sd=tech>

Upgrading to SQL Server 2000 Failover Clustering from an Earlier Version of SQL Server Clustering

In a high availability environment, upgrading your current production SQL Server may affect availability. It is mission-critical to devise a plan that will incur only a minimal amount of downtime. While it is impossible to guarantee 100% uptime during an upgrade, there are options to give you greater amounts of uptime. First, think about these things:

- Is your solution still on the HCL for what you are looking to do? If you are considering not only a SQL Server upgrade, but an operating system upgrade, the solution may have been tested and certified for Windows NT 4.0, but not for Windows 2000.
- If you are using replication, it will have to be disabled/un-configured prior to the upgrade. It would be necessary to have the replication configuration documented, as well as scripted, so that it can be set up after the upgrade. For more upgrade information with replication, see the topics "Backing Up and Restoring Replication Databases", "Scripting Replication", and "Replication and Upgrading" in SQL Server 2000 Books Online.

In conjunction with this, do you also upgrade all of the servers that participate in replication? While it is not necessary, it is always better to maximize your efforts, and minimize downtime by doing all the work in one shot.

- What is your contingency plan? It is critical to have reliable backups of your operating system and SQL Server databases. Ideally, new hardware would be purchased and configured from scratch to minimize risk and downtime to the production environment. In this case, the hardware would be configured, and the application or Web site would only need to incur downtime to upgrade the databases. Downtime may even be eliminated if, for example, the backups from the previous night were used, and then a differential of the data were applied to the server (if possible). Ultimately, your upgrade strategy must take into account the availability of your application with the uptime agreement in place with the users of the system.

Upgrading the Operating System

Windows 2000 Advanced Server and Windows 2000 Datacenter Server offer some new features that enhance SQL

Server 2000, such as AWE Memory and greater scalability (for example, a larger number of processors and a larger base memory). If you are upgrading from Windows NT 4.0, Enterprise Edition, you can upgrade directly to Windows 2000 Advanced Server, but not directly to Windows 2000 Datacenter Server. Consult the Hardware Compatibility List to see whether your current cluster solution and components are certified for use with Windows Clustering.

The upgrade from Windows NT 4.0, Enterprise Edition to Windows 2000 Advanced Server is considered a "rolling upgrade." You do not have to uncluster your failover cluster; instead fail all services over to the other cluster node prior to upgrading the server.

Follow the steps located at <http://www.microsoft.com/windows2000/server/howtobuy/upgrading/path/winnt4ent.asp>

This is a permanent change, and there is no uninstall feature. In a development environment, Windows 2000 Advanced Server can be used to test your application for an eventual Windows 2000 Datacenter Server rollout.

For additional information, see the following sources:

- General upgrade information at <http://www.microsoft.com/windows2000/server/howtobuy/upgrading/default.asp>
- For more detailed information about rolling upgrades, see <http://windows.microsoft.com/windows2000/en/advanced/help/>
- Knowledge Base article "249735 – Error Message Upgrading Server Cluster Node to Windows 2000," at <http://support.microsoft.com/default.aspx?scid=kb;en-us;249735&sd=tech>

Upgrading from SQL Server 6.5

A SQL Server 6.5 cluster and a SQL Server 2000 cluster cannot be configured on the same hardware. Unlike the operating system, you cannot do a rolling upgrade of SQL Server. You will incur some downtime when upgrading. After the version upgrade, it is recommended that all client computers accessing SQL Server 2000 be upgraded to Microsoft Data Access Components (MDAC) version 2.6, which is the version installed on the server.

To upgrade from SQL Server 6.5, consider the following strategies:

- Upgrade on the same hardware

If you are looking to use the same hardware and have determined that it is still on the HCL for the operating system you are using, SQL Server 2000 Books Online has instructions for doing both an active/passive and active/active upgrade. It involves unclustering SQL Server 6.5, installing a local instance of SQL Server 2000, upgrading the data from SQL Server 6.5 using the Upgrade Wizard, and then upgrading the local instance to a clustered instance. During the upgrade, make sure that you use the proper cluster disks for the data. Make sure you have a complete and thorough list of your database server configuration (for example, how to build the segments in your database and where to put the files) so you can rebuild the server if necessary – this would involve reinstalling the operating system and SQL Server 6.5.
- Start with a new Windows 2000 Advanced Server or Windows 2000 Datacenter Server configuration. Install SQL Server 6.5 (for information, see the Knowledge Base article "192710 – Basic Guidelines for Installing SQL Server Version 6.5 or 7.0" at <http://support.microsoft.com/default.aspx?scid=kb;en-us;192710&sd=tech>), restore your 6.5 databases to this new server, and follow the guidelines from the previous bullet point. However, unlike the previous bullet point, this allows a great contingency plan if something happens in the upgrade process because your old hardware is still configured and ready for use.
- Start with a new Windows 2000 Advanced Server or Windows 2000 Datacenter Server configuration, and install a clustered instance of SQL Server 2000. Create a blank database and import the data from SQL Server 6.5 using Data Transformation Services (DTS) or a method such as BULK INSERT or bulk copy (BCP). As with the previous bullet point, this would provide a reliable contingency plan if the data import to SQL Server 2000 does not work. This may also minimize downtime because most of the upgrade is accomplished while the old servers are still in use. This method should be tested to see if it degrades performance or availability.
- Consider upgrading the SQL Server 6.5 database to SQL Server 7.0 as an intermediary step, and then follow the guidelines below for upgrading a SQL Server 7.0 configuration.

Upgrading from SQL Server 7.0

As with SQL Server 6.5, a SQL Server 7.0 cluster and a SQL Server 2000 failover cluster cannot be configured to run on the same hardware, and a rolling upgrade cannot be done with SQL Server – you will incur some downtime. After the version upgrade, it is recommended that all client computers accessing SQL Server 2000 be

upgraded to Microsoft Data Access Components (MDAC) version 2.6, which is the version installed on the server.

To upgrade from SQL Server 7.0, consider the following strategies:

- Use log shipping. This option assumes that the SQL Server 2000 failover cluster is on new hardware. Log shipping can be manually set up from SQL Server 7.0 configured with a minimum of SQL Server 7.0 Service Pack 2 to SQL Server 2000 Enterprise Edition. For more information, see the documentation for the service pack. There are a couple of caveats:
 - On SQL Server 7.0, the "pending upgrade" option must be set to **true** with `sp_dboption`. However, this will mean that users cannot create indexes or statistics in the database, and errors will be generated. It is for this reason that log shipping from SQL Server 7.0 to SQL Server 2000 should be done for a finite amount of time.
 - When restoring the proper point-in-time full database backup to apply the subsequent transaction log backups, restore the database in SQL Server 2000 with `NORECOVERY`.
 - There will be no graphical way to monitor log shipping as there is when it is configured between two SQL Server 2000 Enterprise Edition servers. You will have to query the log shipping tables directly.

Because of these considerations, it is recommended that you not use log shipping between SQL Server 7.0 and SQL Server 2000 for an extended time frame in a production environment.

Log shipping will provide the maximum amount of uptime, as it will allow the current production database to be up and processing requests while you are upgrading the SQL Server 2000 virtual server that will become the new production database. It also provides a contingency plan, as you are not affecting the current production hardware.

To bring the new SQL Server 2000 database online, perform the following:

- At a chosen point of time, access to the current production database should be curtailed.
- Once all the connections are spun down, make sure all the transaction logs are applied to the SQL Server 2000 database.
- Bring the database online with the `WITH RECOVERY` option of the `RESTORE` command in Transact SQL. This can be accomplished on the database after the last transaction log is completed, or expressly put at the end of the last transaction log restore statement.
- Redirect any clients or applications to the new database and server
- Test to ensure that everything is functioning as expected, and open up the database for general use
- Upgrade on the same hardware

The process is more straightforward than the one for SQL Server 6.5. However, you still must check to see if the hardware solution is still on the HCL for the operating system you are using. The instructions for upgrading an active/passive or active/active configuration are in SQL Server 2000 Books Online in the topic "Upgrading to a SQL Server 2000 Failover Cluster." You may have problems if your shared cluster disk is not configured with the proper logical drives for the data.

There is a big caveat if this is your option: MDAC 2.6, which is installed with SQL Server 2000, is incompatible with SQL Server 7.0 clustering. So if you need to revert back to SQL Server 7.0 and MDAC 2.5, test this process in a staging environment, as you have only one chance to rebind in a production environment. Otherwise, a complete server rebuild (from the operating system up) is necessary. For more information, see the Knowledge Base article "239473 PRB: 70rebind.exe for Windows 2000 and MDAC Upgrades on Clustered SQL Server 7.0 Servers" at <http://support.microsoft.com/default.aspx?scid=kb;en-us;239473&sd=tech>

- Start with a new Windows 2000 Advanced Server or Windows 2000 Datacenter Server configuration, and install a clustered instance of SQL Server 2000. Backup and restore the SQL Server 7.0 databases on SQL Server 2000, or use the Copy Database Wizard. This also leaves the SQL Server 7.0 database intact and allows a reliable contingency plan.
- Start with a new Windows 2000 Advanced Server or Windows 2000 Datacenter Server configuration, and install a clustered instance of SQL Server 2000. Create a blank database, and import the data from SQL Server 7.0 using Data Transformation Services (DTS) or a method such as `BULK INSERT` or bulk copy. Because the old servers are still configured, this provides a reliable contingency plan if the data import to SQL Server 2000 does not work. This method also accounts for any changes to files or file group locations.

Upgrade Order

After you understand and evaluate the considerations for upgrading a previous version of SQL Server clustering, the next question is, in what order should the upgrade take place? If you are doing both an operating system and a SQL Server upgrade on the same hardware, perform the operating system first as you can do a rolling upgrade and still service requests in your database. In the upgrade from Windows NT 4.0 to Windows 2000, make sure that attention is paid to detail, such as running comclust.exe on each node to ensure that MS DTC is clustered. Even if you are using the same hardware and staying with your current operating system, ensure it meets the requirements (such as service pack levels) for SQL Server 2000 prior to upgrading SQL Server. Upgrading to SQL Server 2000 should be the last step, including restoring any custom configurations (like replication).

Important: Before upgrading your production databases, test the process in a testing/staging environment. You want to make sure you know any potential problems you may encounter prior to rolling out the upgrade in a production environment, because in a high availability environment, downtime is crucial – each minute counts. One of the biggest things to know is how long the upgrade will take. Even if you cannot test on clustered machines, it is better to test the upgrade on standalone machines than to not test at all.

Upgrading a Standalone (Local) SQL Server 2000 Instance to a SQL Server 2000 Virtual Server

Upgrading from a local instance to a clustered instance is possible. However, there are a couple of things to take into consideration:

- If your hardware is not configured as a cluster, you cannot turn it into a cluster. As stated earlier in this paper (see the section "Hardware Compatibility List"), the solution must be purchased as a cluster.
- If your data is not on the proper cluster disks, or the drive array is not configured properly, you will have the same problems as a SQL Server 6.5 and 7.0 upgrade. You must make sure that the data is on the proper drives prior to installation, or have a way of moving them by using a method such as attach and detach.

If these considerations are not a problem, make sure Windows Clustering is configured, and then upgrade the instance to a clustered instance. For a sample of how to upgrade, see the topic "How to upgrade from a default instance to a default clustered instance of SQL Server 2000 (Setup)" in SQL Server Books Online.

[↑ Top of page](#)

Verifying Your Failover Cluster Installation

This section provides some methods for testing whether your failover cluster is configured properly. Perform these tests prior to putting the failover cluster into production. This is recommended to ensure client connectivity and to prevent an interruption in availability due to testing. For some checklists to use, as well as more information, see Appendix D "Pre- and Post-Installation Checklists" at the end of this paper.

Verifying Connectivity and Name Resolution

To verify that the private and public networks are communicating properly, perform the following steps. It is imperative to know the IP address for each network adapter in the cluster, as well as for the IP cluster resources (which can be entered on the worksheet found in Appendix C in this paper).

Verifying Connectivity and Name Resolution from a Server Node

This method shows how to check both IP connectivity and name resolution at the server level.

1. On a node, on the **Start** menu, click **Run**, and then type **cmd** in the text box. Click **OK**.
2. Type **pingipaddress/servername** where *ipaddress/servername* is the IP address for the corresponding network adapter in the other node or the virtual server name for your cluster.
3. Repeat for each node.

For example, assume that the IP addresses are set as shown in the following table.

Node	Item to check	Value
1	Public Cluster Connection	199.1.3.16
1	Private Cluster Connection	10.1.1.1
2	Public Cluster Connection	193.1.2.199
2	Private Cluster Connection	10.1.1.2

Node	Item to check	Value
N/A	Cluster IP Address	199.2.6.4
N/A	SQL Server IP Address	199.10.3.6
N/A	Cluster Name	MyCluster

In this example, you would type **ping 193.1.2.199** and **ping 10.1.1.2** from Node 1, and you would type **ping 199.1.3.16** and **ping 10.1.1.1** from Node 2. Both nodes should also be able to ping the Cluster IP Address, the Cluster Name, as well as the SQL Server IP Address.

Verifying Connectivity and Name Resolution from a Client

This method shows how to check both IP connectivity and name resolution at the client.

1. On a client computer, on the **Start** menu, click **Run**, and then type **cmd** in the text box. Click **OK**.
2. Type **ping ipaddress/servername** where *ipaddress/servername* is the IP address for the corresponding network adapter in the other node or the virtual server name for your cluster.

In this example, if you follow the previous chart, type **ping 193.2.6.4** and **ping MyCluster** to check connectivity to the cluster. To check the IP address for SQL Server, type **ping 199.10.3.6**. To check your connection to SQL Server using a named connection or IP, set up an ODBC connection or use the Client Network Utility.

Failover Validation

Finally, perform a failover of all SQL Server virtual servers to ensure that all resources fail over and restart on another node with no problems or affecting any other groups. The exception to this would be if SQL Server used the quorum drive against previous recommendations. This is achieved using Cluster Administrator.

1. On the **Start** menu, point to **Programs** and **Administrative Tools**, and click **Cluster Administrator**.

All nodes configured for the failover cluster should appear in the bottom of the left pane of Cluster Administrator.

2. Right-click a group containing the SQL Server resources (for example, **SQL Server Ins1**), and then click **Move**. The group selected and its resources will now be moved to its preferred failover node. This change will be reflected in Cluster Administrator.

Verifying the Service Account

For SQL Server to be able to manage its resources and perform correctly the service account must be part of the cluster ACL. To ensure that this is configured properly, execute the following in a SQL Query Analyzer window:

```
select * from ::fn_virtualservernodes()
```

If there is no output, ensure the account SQL Server is running under is part of the cluster ACL.

[↑ Top of page](#)

Maintaining a SQL Server 2000 Failover Cluster

Maintaining a failover cluster can be challenging. For example, how do you create an environment that is seamless and works no matter what node owns the SQL Server processes? This section presents some unique considerations you be aware of when maintaining a clustered environment.

Administering SQL Server Virtual Servers

There are four places you can administer your SQL Server 2000 virtual server. It is important to understand the similarities and differences between them so you use the right tool.

- SQL Server tools, especially SQL Server Enterprise Manager

SQL Server Enterprise Manager and the other SQL Server tools should be used to administer the database. All accounts and passwords associated with SQL Server and SQL Server Agent should be changed in Enterprise Manager, and if a port number needs to be changed, use Server Network Utility. Use the other SQL Server tools as you would for a non-clustered instance.

- SQL Server Setup

To uninstall the virtual server, add or remove the nodes participating in the failover cluster, or to change or add IP addresses to the failover cluster, use SQL Server setup.

- Cluster Administrator

This tool is an operating system-level tool, and is located in Administrative Tools. Prior to SQL Server 2000, most configuration changes to SQL Server clustering were done in Cluster Administrator. Only use Cluster Administrator where it is outlined in this paper to ensure proper use with SQL Server 2000 failover clustering. Do not use Cluster Administrator to add nodes to the resource definitions or modify IP addresses.

- The command line CLUSTER utility

The CLUSTER command line tool is basically the operating system command-line interface for most functionality within Cluster Administrator. As with Cluster Administrator, only use it when necessary.

Windows 2000 Datacenter Process Control and SQL Server 2000 Failover Clustering

Important: Do not use Process Control to modify SQL Server virtual server configurations. Process Control is not a cluster-aware application, and in the event of a failover, the virtual server modified on one node will not carry over the process control constraints from the failed node automatically. Use SQL Server Enterprise Manager and the other SQL Server-supplied tools to modify the SQL Server virtual server configuration.

Backing Up and Restoring

Although backing up your databases in a clustered environment is not completely unlike that of a normal server, it definitely is more complex. So how do you handle such a situation? When a system like a cluster is configured, it is used for large and mission-critical databases. Backing up and restoring databases in the terabyte range cannot be handled as you would a 10 MB database, although many try to treat it as such. General best practices apply:

- Make frequent backups.
- Have offline storage rotation for backup files put onto tape or any other media.
- Test and time restores on all backups so in the event of an emergency, you not only know the backup is good, but how long it takes. Knowing this is crucial in some server down situations.

Important: Do not use the quorum drive to store backups.

Backing Up to Disk and Tape

More often than not, it is easiest to first back up to disk. Create a cluster disk share so in the event of a fail over, all nodes will have access to the backup share. Do not attempt to back up to any local drives. After the database is backed up, it should be copied to another location, backed up to another medium such as tape, and then archived in an offsite location after it is tested and verified. The goal of backups in a high availability environment is to remove a single point of failure, so if you make a backup and just keep it on a drive somewhere, even if RAID is used, what do you do if the array fails? While this is unlikely, the worst-case scenario must be considered.

Another way is to provide two steps in the backup job. Set up two backup methods (for example, tape drive and a shared cluster disk). Set up your maintenance plan, and then alter the backup task. If the backup succeeds in step one, exit with success, but if it fails (for any reason), the second method is invoked. This would ensure that there is not a single point of failure in your backup strategy.

Snapshot Backups

One way to back up and restore a clustered SQL Server is to use a snapshot backup. SQL Server 2000 supports snapshot backups, which involves the mirroring of your disks, breaking a complete set of disks off the mirror, and then using them as a backup. Snapshot backups require specialized hardware; and are supported by SQL Server 2000.

Example

TerraServer (<http://www.teraserver.com/default.asp>) is a Web site that delivers aerial photographs and maps of geographic locations provided by the United States Geological Survey. The database is currently nearing two terabytes of data, and it uses the Windows 2000 Datacenter Server with SQL Server 2000 failover clustering in the N+1 scenario. As you can imagine, backing up this very large database (VLDB) is something that had to be planned carefully.

TerraServer employs a snapshot backup. They have three disk mirrors in addition to RAID (think of them as three

columns lined up next to each other), that is, three copies of the data are kept in synchronization by hardware. Therefore, in the event of a disk failure, you have two sets of backups. However, at some point, one of the mirror sets is broken off, and essentially, it becomes a live spinning backup of the database. However, at the time you break it off, it will no longer be kept synchronized, nor will SQL Server see it. SQL Server 2000 is smart enough to react, and handles its memory buffers appropriately. They then use a tape solution to back this live volume up, and at some point, spin the disk set back in, and you are back at three mirrors. This happens on a cyclical basis.

Backing Up an Entire Clustered System

It is not enough to just back up your SQL Server 2000 databases. You must also back up your complete system. It is also important to back up the system state of a Windows cluster, and if it needs to be restored, restore the system state after the operating system is put on the machine. This requires a cluster-aware backup program. Some third-party vendors can provide this service.

For native tools, consider the following:

- **ntbackup.exe.** This tool backs up and restores cluster configuration, which includes the quorum disk and system state. The tool does not work with remote servers. If the server is running the Cluster service, the System State data will also include any resource registry checkpoints and the quorum resource recovery log, which contains the most recent cluster database information.
- **clusrest.exe.** This tool restores the contents of backup quorum log to live quorum.
- **clustool.exe** This tool backs up and restores certain parts of the cluster configuration. It also includes a migration tool for porting stand-alone file and printer shares to a cluster. Core resources such as the cluster IP address, cluster names, quorum disks are not restored. This tool is available from the Windows 2000 Server Resource Kit (\apps\clustool\). It replaces clusconb.exe.
- **dumpcfg.exe** This tool backs up and restores disk signatures. It is installed as part of the Windows 2000 Server Resource Kit.
- **Cluster Automation Server.** This series of ActiveX® controls for working with Cluster service is part of Windows 2000 (msclus.dll). If you are running Windows NT 4.0, it is available on the Windows 2000 SDK CD (Redist\Cluster\NT4\i386).

The previous considerations for backing up to disk and tape still apply: Ensure that there are no single points of failure and all nodes have access to the same devices in the same way.

Ensuring a Virtual Server Will Not Fail due to Other Service Failures

To prevent the failure of specific services from causing the SQL Server group to fail over, configure those services properly using Cluster Administrator. See step 4 of the "To configure thresholds for a resource" section earlier in this paper for instructions. For example, if SQL Server Full-Text functionality is not used as part of your solution, you should ensure that the **Affect the group** parameter is deselected from the properties of the resource.

Adding, Changing, or Updating a TCP/IP Address

Until SQL Server 2000, changing the TCP/IP address required SQL Server to be unclustered if SQL Server clustering was implemented. To change the TCP/IP address in SQL Server 2000, run the Setup program again. In addition, due to the new multiple network card/IP address support built into SQL Server 2000, additional TCP/IP addresses can be configured for the instance. However, you are limited to one IP address per subnet. For example, if you have internal and external customers accessing the instance, you can assign SQL Server two separate IP addresses to maximize network use and to simplify tracking the use of your instance of SQL Server.

To add, change, or update a TCP/IP address

1. Insert the SQL Server 2000 Enterprise Edition compact disc in your CD-ROM drive. Select **Install SQL Server 2000 Components**.
2. Click **Install SQL Server 2000 Components**, click **Install Database Server**, and then click **Next**.
3. On the **Computer Name** dialog box, select **Virtual Server**, and enter the name of an existing clustered instance of SQL Server 2000.
4. On the **Installation Selection** dialog box, select **Advanced options**, and then click **Next**.
5. On the **Advanced Options** dialog box, select **Maintain a Virtual Server for Failover Clustering**, and then click **Next**.

6. On the **Failover Clustering** dialog box, a TCP/IP address can be added to or removed from the selected instance of SQL Server 2000.

To remove a TCP/IP address, select the address, and click **Remove**.

Note: An instance of SQL Server 2000 in a failover cluster requires a TCP/IP address to function. Only remove a TCP/IP address if more than one exists and if this will not affect users or applications accessing SQL Server.

To add a TCP/IP address, enter the new TCP/IP address in the **IP Address** box, select the network to use, and then click **Add**. The new IP address appears after the existing IP address.

7. On the **Cluster Management** dialog box, click **Next**.
8. On the **Remote Information** dialog box, enter the user name and password for the domain administrator account used for the clustered instance of SQL Server 2000, and then click **Next**.
9. When the process is complete, click **Finish**.

Adding or Removing a Cluster Node from the Virtual Server Definition

Another new feature of SQL Server 2000 failover clustering is the ability to add or remove a cluster node from a SQL Server virtual server definition. Adding nodes to the existing SQL Server virtual server definition performs all the necessary operations on the new nodes (including installing binaries, system components, and creating services) and performs the necessary modifications to the cluster configuration.

To add or remove a node

1. Insert the SQL Server 2000 Enterprise Edition compact disc in your CD-ROM drive. Select **Install SQL Server 2000 Components**.
2. Click **Install SQL Server 2000 Components**, click **Install Database Server**, and then click **Next**.
3. On the **Computer Name** dialog box, select **Virtual Server**, and enter the name of an existing clustered instance of SQL Server 2000.
4. On the **Installation Selection** dialog box, select **Advanced options**, and then click **Next**.
5. On the **Advanced Options** dialog box, select **Maintain a Virtual Server for Failover Clustering**, and then click **Next**.
6. On the **Failover Clustering** dialog box, click **Next**.
7. On the **Cluster Management** dialog box, select the appropriate nodes to add or remove from the cluster, and then click **Next** when you are finished.
8. On the **Remote Information** dialog box, enter the user name and password for the domain administrator account used for the clustered instance of SQL Server 2000, and then click **Next**.
9. When the process is complete, click **Finish**.

Renaming a SQL Server 2000 Virtual Server

Renaming a SQL Server 2000 virtual server is neither possible nor supported. The only way to remove a virtual server is to uninstall it.

Applying a SQL Server 2000 Service Pack

As mentioned earlier, SQL Server 2000 no longer requires you to uncluster your clustered SQL Server implementation to apply a service pack. It is recommended that you review the readme file before installing a service pack, as it will contain information specific to that release. For example, SQL Server 2000 Service Pack 1 requires a reboot after installation, which will affect availability. In addition, before installing a service pack to a virtual server, consider the following:

- Upgrading a virtual server with a service pack is the same as upgrading a single instance of SQL Server. You will need to repeat the installation of the service pack for each virtual server in your Windows cluster. Installer will upgrade the underlying components on all nodes that are part of the virtual server definition.
- The failover cluster resources for the selected virtual server must be online and running for a successful service pack installation.

- During the upgrade, the selected virtual server will be unavailable to client requests. The service pack may also require that the failover cluster nodes be rebooted. Plan for this interruption of availability, and let your end users know in advance so they can plan accordingly.
- Check the readme to see which components are upgraded in the service pack. However, if MS DTC is one of those components, and there is more than one virtual server using MS DTC in your cluster, the other virtual servers may be affected during the upgrade process to the selected virtual server since it is a shared resource in the cluster.
- Prior to installing the service pack, back up all system and user databases, and ensure that the system databases have enough free space.
- To revert to the version of SQL Server installed prior to the service pack installation, you will need to uninstall your virtual server, reinstall SQL Server 2000, and then recreate your user databases by attaching or restoring.

[↑ Top of page](#)

Troubleshooting SQL Server 2000 Failover Clusters

Troubleshooting SQL Server 2000 in a failover cluster configuration is not always the same as diagnosing problems on a standalone server. First, you need to verify that the hardware, operating system, and Windows Clustering are all functioning properly. Then, if all of those factors are in good health, turn to SQL Server. For more information, see "Failover Cluster Troubleshooting" in SQL Server Books Online

Service Level Agreements

Ensure that you have a Service Level Agreement (SLA) with your hardware and software vendors that match the level of support you expect. Because a failover cluster is usually a mission-critical production system, purchasing a SLA that guarantees a 48-hour turnaround may not be effective. The value of a support contract may be the difference between a small amount of downtime, and a very long amount of downtime. It is imperative in a production environment that a support call is placed prior to any troubleshooting, because troubleshooting may increase downtime if it is a server down situation.

First Steps

Check the operating system's Event Viewer at the Application, System, and Security logs. Sometimes the problem is apparent, such as a disk or network card failure, or relevant error messages from the operating system or SQL Server may appear. Next, check the cluster logs, which are located where the system variable `%clusterlog%` is set (generally `\\winnt\\cluster`). These are the files:

- `Sqlstpn.log`. The log for the SQL Server Setup, where *n* is the number of the setup attempts.
- `Sqlclstr.log`. The log for the clustered instances of SQL Server.
- `Cluster.log`. The main cluster log file.

For complete instructions for enabling and disabling logging, see the Knowledge Base article 168801 "How to Enable Cluster Logging in Microsoft Cluster Server", located at <http://support.microsoft.com/default.aspx?scid=kb;en-us:168801&sd=tech>

This information is also invaluable if a call is placed to your hardware vendor or Microsoft Product Support Services. The more information you can provide, the more quickly they will be able to help you resolve your issue.

Repairing a Single Node Failure and Quorum Disk Failure

If a single node fails in the cluster (due to hardware failure) or the quorum disk fails, follow these steps to rebuild the node and rejoin the cluster:

1. After verifying that all cluster resource groups have been successfully moved to another node in the cluster, use the SQL Server installation program to remove the node from the SQL Server virtual server definition (see the section "Adding or Removing a Cluster Node from the Virtual Server Definition" earlier in this paper for instructions). Setup should detect the lost node and automatically move it to the **Unavailable nodes** list.
2. Evict the server node using Cluster Administrator.
3. Repair the cluster node. This may require building a new cluster node or restoring from a recent backup.
4. Rejoin the cluster.

5. If MS DTC is used, run comclust.exe on the node.
6. Rerun the SQL Server 2000 install on the node that did not have the problem. Select the evicted node from **Available nodes** list and add it to the **Configured nodes** list.

Note: Evicting a node from the cluster definition prior to removing it from the SQL Server virtual server definition will cause problems. If a node is evicted, it will not be displayed in SQL Server Setup, and thus, will not be cleaned up properly. This can cause the cluster to inappropriately add the node as a possible owner for SQL Server resources after it rejoins the cluster if the SQL Server resource DLL is left in place on the evicted node. This is a rare occurrence, but you should consider it. Also, if you attempt to add the node back without first removing it from the Windows cluster definition, SQL Server 2000 failover clustering may become broken.

Multiple Node Failure

If more than one node fails, but not all nodes fail, repeat the preceding steps to repair all nodes. However, if all nodes fail, and the quorum cannot be repaired, you will have to rebuild all nodes in the cluster. This is why tested and frequent backups are crucial.

Rebuilding the master Database in a Clustered Environment

If the **master** database needs to be rebuilt in a SQL Server 2000 failover cluster, follow these steps:

1. Go to the node currently owning the SQL Server resource.
2. Bring the SQL Server virtual server offline using Cluster Administrator.
3. Make sure the original shared installation files or the SQL Server CD are available.
4. If you are using the SQL Server CD, copy all files from the CD to a local hard drive. Remove the read-only attribute from the files after copying them to the hard drive.
5. Execute rebuildm.exe. Point it to the original shared installation files or to the files copied from the CD to the local hard drive.
6. Choose Windows or SQL Collation.
7. After rebuildm.exe completes, verify that the resources can be brought online and they successfully fail over.
8. Verify the collation by executing **sp_helpsort**.

These steps do not include the steps necessary to deal with user databases. In this case, SQL Server 2000 was a new installation and included only the databases shipped with SQL Server. If you have a recent backup of **master**, you may be able to restore it at this point. If you do not, you will have to restore or attach the user databases.

Common Troubleshooting Issues

This section presents some of the common issues and solutions when implementing failover clustering.

Q: When the installation process attempts to install the SQL Server binaries on the other node(s), it fails (possibly with a failed logon request error). Why?

A: If you are installing from a network share, make sure that both nodes have connectivity to the share without specifying a network password to connect (for example, you should be able to view `\\sharecomputer\sharepath` without specifying credentials). If you are installing from CD-ROM drive on Node A, make sure the cluster nodes are configured to communicate properly, and that the proper accounts exist on each node, and that the other nodes are set up for Windows Authentication. Mapping a drive letter, even if it is the same on all nodes, will not work, because the installation process accesses the UNC path of the share.

Q: After installing and rebooting the server, the SQL Server install doesn't seem to complete. Why?

A: Sometimes file renames are blocked due to locks on startup (for example, MDAC); therefore, if the file remains read only, it never completes.

Q: MSCS lost connectivity to my SQL Server virtual server. Why?

A: This may be because the process used to perform the IsAlive check is run in the context of the MSCS service account. This account must have **sysadmin** rights to SQL Server. If this behavior is not exhibited, check all logins and cross-reference with the cluster logs to see if there is an IsAlive check.

Q: Problems ensue when changing the network name of SQL Server after the install. How can this be fixed?

A: SQL Server binds itself to the network name used at install time. If it is changed for some reason, in certain occurrences, it may require a complete reinstall.

Q: After an installation of a new SQL Server virtual server, clients cannot connect to the server, especially using graphical user interface (GUI) tools. Why?

A: The DNS and WINS servers need to refresh to recognize the new SQL Server 2000 virtual server installation, which may take a little bit of time. In some cases, it may be necessary to manually insert the entries into the DNS and WINS configuration files.

Q: Microsoft Exchange 2000 and SQL Server 2000 are both installed on my cluster; however, the full-text service in SQL Server seems to be failing. Why?

A: If both must exist on the same cluster (although this is not recommended), install Exchange 2000 first, and then install SQL Server 2000.

Q: I am encountering problems with full-text setup in my failover clustering. What might be wrong?

A: In some cases, Setup may fail because the underlying Microsoft Search service does not exist. If this is the case, you may have to manually create the type to get the installation to complete next time. To create the resource, execute the following from a command prompt:

```
cd %windir%\cluster
regsvr32 gathercl.dll
```

Ensure that the file is registered on all nodes. You can then rerun Setup and everything should work fine.

[↑ Top of page](#)

Conclusion

SQL Server 2000 Failover Clustering is the leading option as a primary method to achieve high availability for your databases. It offers full transactional consistency, and a failover to other another node that is automatic. By eliminating single points of failure both at the software and at the hardware level and with the proper processes and disaster recovery plans in place, you may be able to achieve five nines of availability.

[↑ Top of page](#)

Appendix A –Additional Information

For more information about SQL Server 2000 failover clustering, see the following resources:

- Microsoft SQL Server home page at <http://www.microsoft.com/sql/>
- For more information about the architecture, technology, and terminology of a Windows cluster, see "Windows Clustering Technologies: Cluster Service Architecture," at <http://www.microsoft.com/windows2000/techninfo/howitworks/cluster/clusterarch.asp>
- For more information about failover clustering and SQL Server 2000 in general, see SQL Server 2000 Books Online.
- For more information about clustering in Windows NT 4.0, Enterprise Edition Server at <http://www.microsoft.com/ntserver/ProductInfo/Enterprise/default.asp> and in Windows 2000 Advanced Server and Datacenter Server at <http://www.microsoft.com/windows2000/technologies/clustering/default.asp>
- High Availability Operations Guide for Windows NT Server 4.0 at <http://www.microsoft.com/ntserver/techresources/deployment/NTserver/highavail1.asp>
- For more information about BizTalk and clustering, see the whitepaper "Deploying BizTalk Server: Clustering Considerations" at <http://msdn2.microsoft.com/library/ms942879.aspx>
- MSDN® developer program at <http://msdn2.microsoft.com/default.aspx>
- For more information on SQL Server 2000 capacity planning, see the *Microsoft SQL Server 2000 Administrator's Companion* at <http://www.microsoft.com/mspress/books/4519.asp>

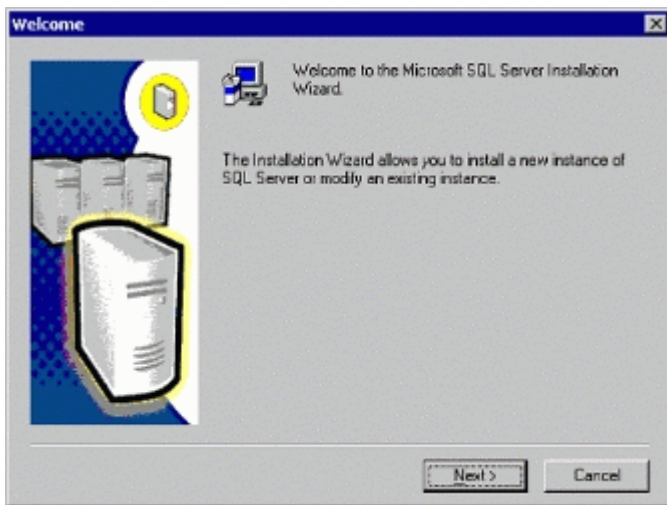
- Microsoft Support Services Web site (contains technical articles and downloadable updates) at <http://support.microsoft.com>

[↑ Top of page](#)

Appendix B – Step-By-Step Installation Instructions for a New Virtual Server

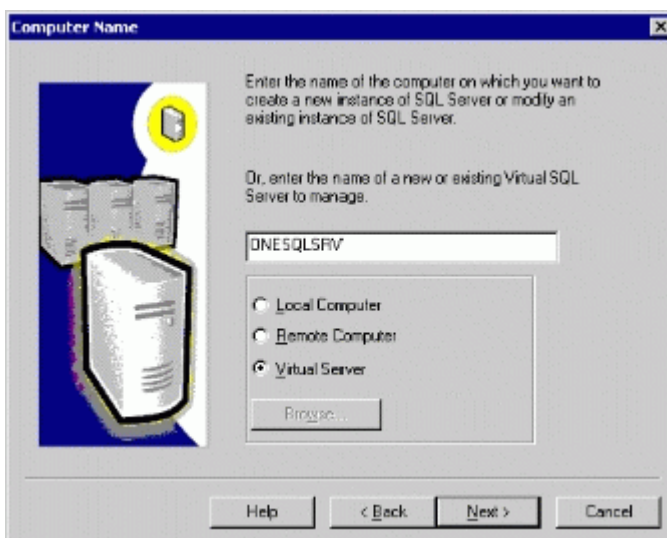
This section will show the installation of a new SQL Server 2000 virtual server.

1. Shut down any unnecessary services that may interfere with the installation process, as defined in the module. If you would like, create two batch scripts that will both start and stop the services.
2. Insert the SQL Server 2000 Enterprise Edition into the CD-ROM drive.
3. When the CD menu appears, click **SQL Server 2000 Components**, and then click **Install Database Server**.
4. In the **Welcome** dialog box, click **Next**.



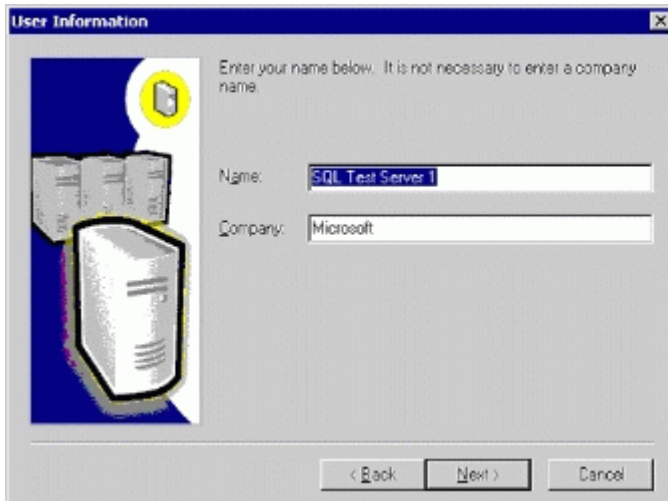
[See full-sized image.](#)

5. In the **Computer Name** dialog box, click **Virtual Server**, type the Network name of your cluster's virtual machine, and then click **Next**.



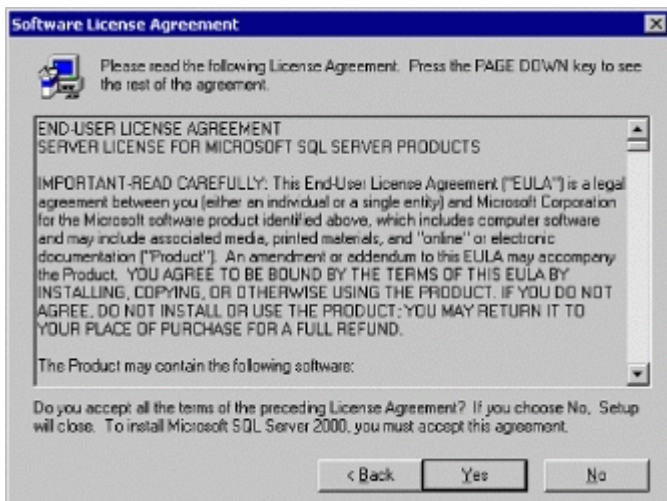
[See full-sized image.](#)

6. In the **Name** box of the **User Information** dialog box, type your name, and in the **Company** box, enter your company. Click **Next**.



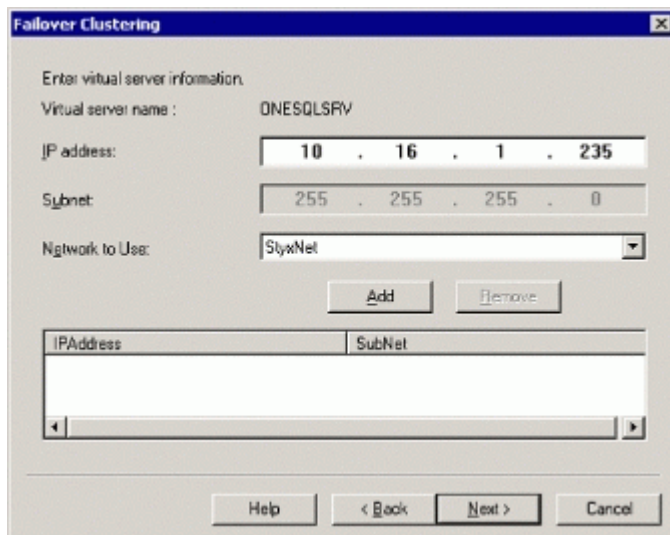
[See full-sized image.](#)

7. In the **Software License Agreement** dialog box, review the **End-User License Agreement** and click **Yes**.



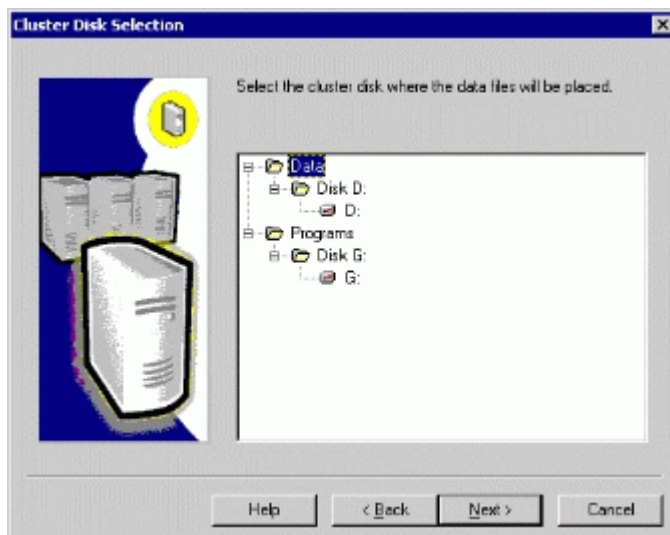
[See full-sized image.](#)

8. Enter the **IP address** that will be used for the virtual server, and select the cluster network to be used from the **Network to use** dropdown list. When complete, click **Add**. Your entries will be displayed in the window at the bottom of the window. You may add more than one IP address at this time. Click **Next** to continue.



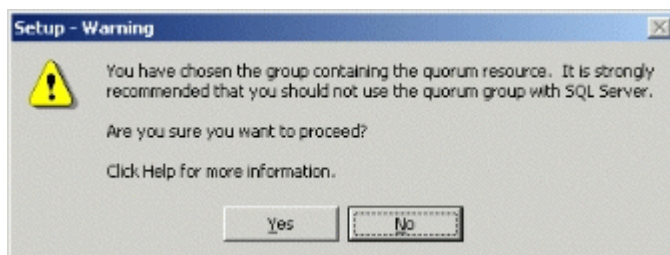
[See full-sized image.](#)

9. In the **Cluster Disk Selection** dialog box, select one drive to place the data files, Adding additional drives must be done post-installation, as noted in the section "Adding A Logical Disk To Your Cluster Configuration" earlier in this paper. Click **Next** to continue.



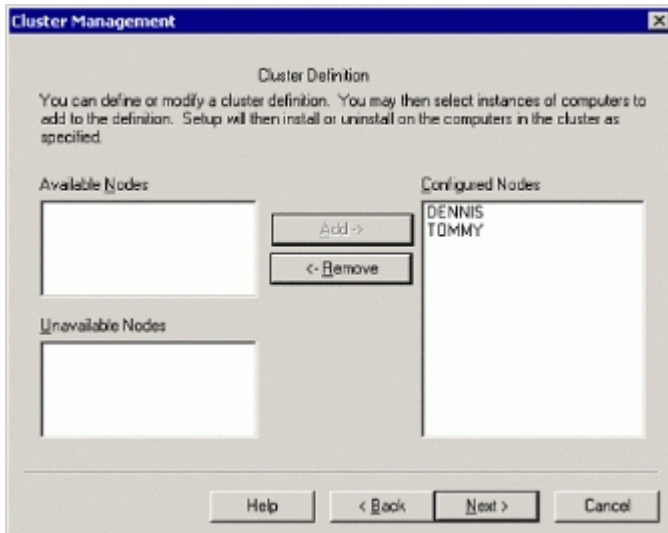
[See full-sized image.](#)

If you are configuring a one-node cluster with the "-localquorum" switch, the only drive that will be available to you is the quorum, and you will see a message similar to the one below. In a production environment, do not use the quorum disk for data or log purposes.



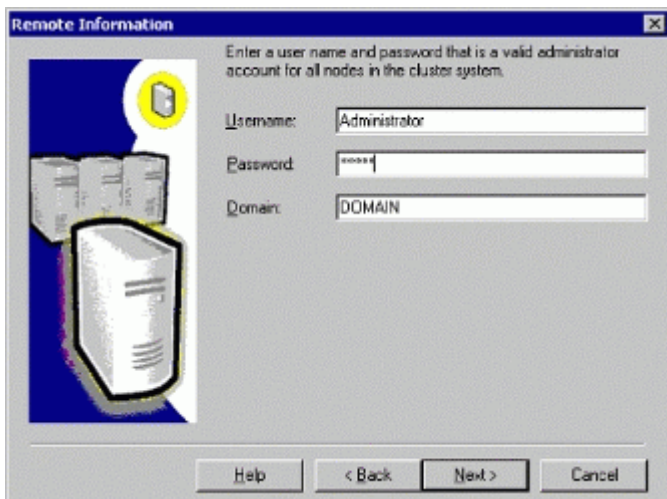
[See full-sized image.](#)

10. In the **Cluster Management** dialog box, you can add or delete cluster nodes from the virtual server definition. Click **Next**.



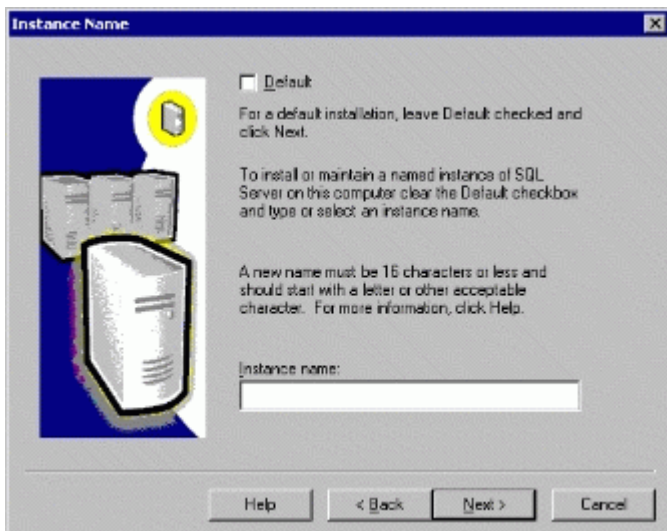
[See full-sized image.](#)

11. In the **Remote Information** dialog box, enter account information used to configure and administer the server cluster. Click **Next**.



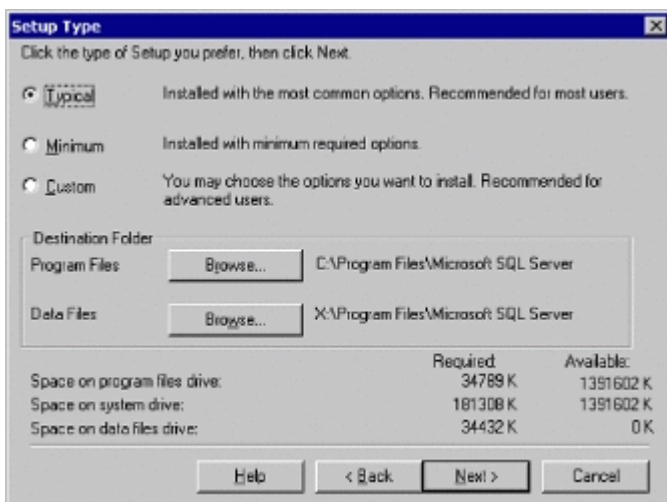
[See full-sized image.](#)

12. In the **Instance Name** dialog box, click **Default** to make the installation the default instance, or leave it unchecked and enter a distinct name for the instance in the **Instance name** entry box. Click **Next** to continue.



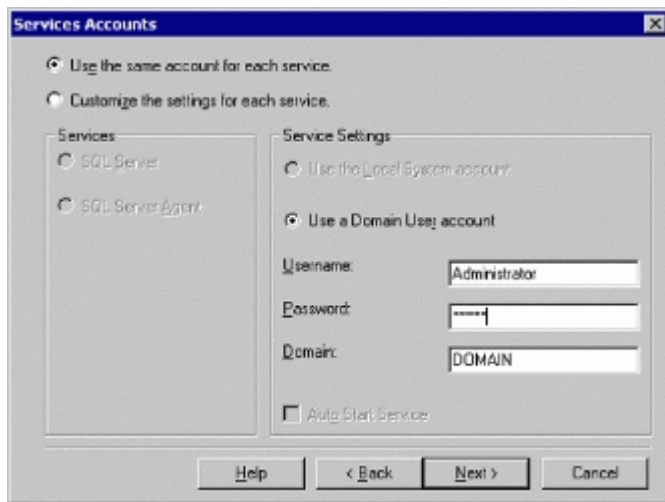
[See full-sized image.](#)

13. In the **Destination Folder** area of the **Setup Type** dialog box, verify that the **Program Files** location is set to a valid local drive on each node (for example, **C:\Program Files\Microsoft SQL Server**) and that the **Data Files** location is set to the drive selected on the **Cluster Disk Selection** dialog box. If you want to configure a specific directory on the drive for the data, click the **Browse** button. Click **Next**.



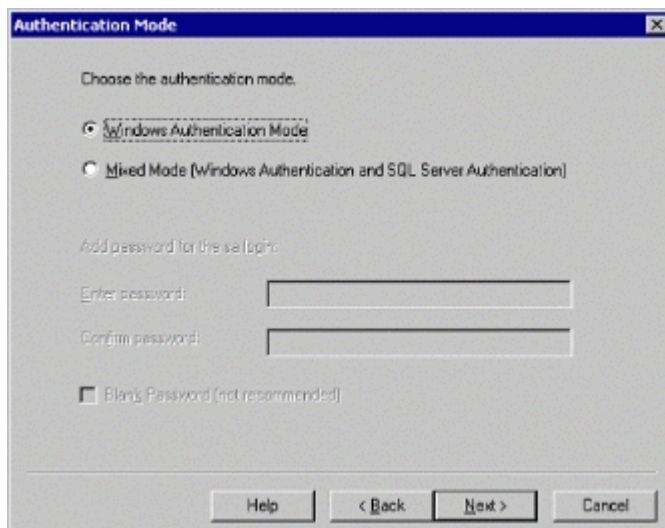
[See full-sized image.](#)

14. In the **Service Accounts** dialog box, select either **Use the same account for each service** or **Customize the settings for each service**. In the **Password** box type **password**. Confirm the **Username**, **Password**, and **Domain** are set to the proper values. If you selected **Customize the settings for each service**, you will need to enter a **Username**, **Password**, and **Domain** for both the SQL Server and SQL Server Agent services. Click **Next**.



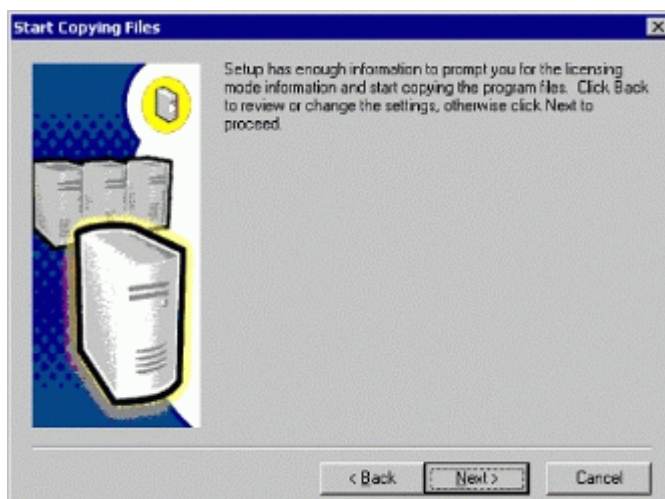
[See full-sized image.](#)

15. In the **Authentication Mode** dialog box, select either **Windows Authentication Mode** or **Mixed Mode**. If **Mixed Mode** is selected, enter a password for the sa account. Click **Next**.



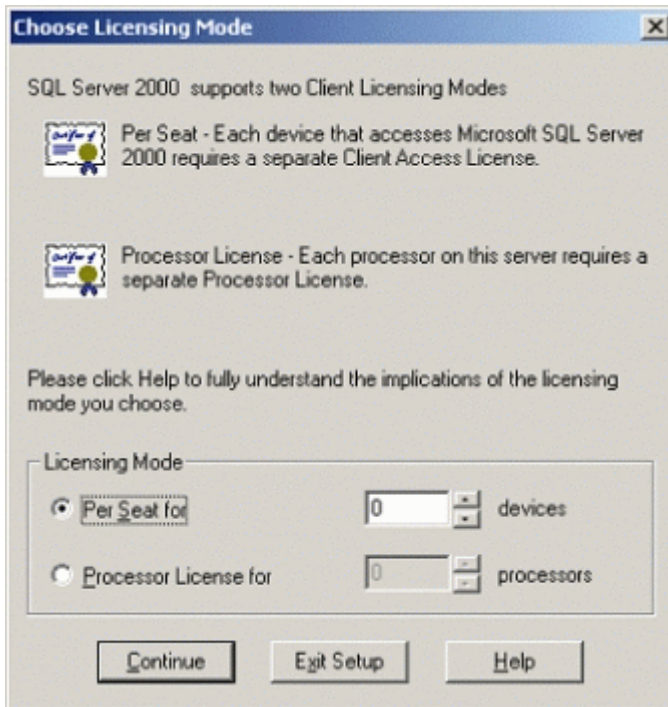
[See full-sized image.](#)

16. In the **Start Copying Files** dialog box, click **Next**.



[See full-sized image.](#)

17. In the **Choose Licensing Mode** dialog box, choose the right licensing scheme, enter the proper value, and click **Continue**.



[See full-sized image.](#)

18. A setup notification box will now appear as the SQL Server 2000 virtual server is installed.



[See full-sized image.](#)

19. In the **Setup Complete** dialog box, click **Finish**. If prompted to restart your servers, make sure you reboot all nodes of your cluster.



[See full-sized image.](#)[↑ Top of page](#)

Appendix C - Configuration Worksheets

Server Cluster Configuration Worksheet

A server cluster should be installed with a minimum of two nodes, and up to four nodes (Windows 2000 Datacenter Server only). When configuring the server cluster using the Cluster Service Configuration Wizard, use this worksheet in conjunction with the Server Cluster Configuration Worksheet.

Parameter	Value
Cluster Name	
Cluster Domain Administrator Account	
Cluster Domain Administrator Password	
DNS Server(s)	
WINS Server	
Cluster IP Address	
Cluster IP Subnet	
MS DTC Network Name (NT 4)	
MS DTC IP Address (NT 4)	
MS DTC Subnet (NT 4)	
Quorum Drive Letter	
Number of Nodes	
Drives For Cluster Use	See "SQL Server 2000 Failover Cluster Disk Configuration Worksheet"

[See full-sized image.](#)

Node Configuration Worksheet

This worksheet will be used to configure each individual server prior to clustering. When configuring the server cluster using the Cluster Service Configuration Wizard, use this worksheet in conjunction with the Server Cluster Configuration Worksheet.

Parameter	Value
Node Number	
Memory	
# of Processors	
Server Name	
Public IP Address 1	
Public IP 1 Subnet	
Public IP 1 (for Cluster Network Configuration Wizard)	Network Name: <input type="checkbox"/> Client access only (public network) <input type="checkbox"/> All communications (private and public networks)
Speed of NIC for Public IP 1	10 <input type="checkbox"/> 100 <input type="checkbox"/> Other <input type="checkbox"/> Value:
Public IP Address 2	
Public IP 2 Subnet	
Public IP 2 (for Cluster Network Configuration Wizard)	Network Name: <input type="checkbox"/> Client access only (public network) <input type="checkbox"/> All communications (private and public networks)
Speed of NIC for Public IP 2	10 <input type="checkbox"/> 100 <input type="checkbox"/> Other <input type="checkbox"/> Value:
Private IP Address	
Private IP Subnet	
Private IP (for Cluster Network Configuration Wizard)	Network Name: Internal cluster communications only (private network)
Speed of NIC for Private Network	10 <input type="checkbox"/> 100 <input type="checkbox"/> Other <input type="checkbox"/> Value:

[See full-sized image.](#)

SQL Server 2000 Virtual Server Configuration Worksheet

Currently, up to 16 instances (1 default, 15 named or 16 named) can be installed on a server cluster. The information on this worksheet is in the order in which it appears during the setup process.

Parameter	Value
Instance #	
Virtual Server Name	
IP Address 1	
Network to Use (IP 1)	
IP 2 Subnet	
Network to Use (IP 2)	
Data/Log Disks Allocated (one used during setup; others added as dependencies after the install)	
Cluster Definition/Preferred Owner Order (nodes part of the virtual server definition)	
Cluster Administrator Information	Username: Password:
Type Of Instance	Default <input type="checkbox"/> Named <input type="checkbox"/> If Named, Name of Instance:
Program Files Location (same location on internal drive on all nodes)	
SQL Server Administrator Account	
SQL Server Administrator Password	
SQL Server Agent Administrator Account	
SQL Server Agent Administrator Password	
Authentication Mode	Windows <input type="checkbox"/> Mixed <input type="checkbox"/> If Mixed, SA password:
Licensing Mode	Per Seat <input type="checkbox"/> Per Processor <input type="checkbox"/>

[See full-sized image.](#)

Shared Cluster Disk Partition Configuration Worksheet

This worksheet will help you configure your shared cluster disk array at the hardware level. Please note that the terminology that applies to the disk array may vary from manufacturer to manufacturer, but the concepts are generally the same.

RAID Partition is a logical grouping of disks in the shared cluster disk array. **RAID Configuration** is the type of RAID configuration that will be used when configuring the disks for use. **Type of Disk** is the basic drive size for each drive that will be part of the RAID Partition. **Number of Disks** is the number of physical drives that make up the RAID Partition. **Partition Size** is the amount of space available to the operating system for use.

Example:

RAID Partition	RAID Configuration	Type of Disk	Number of Disks	Partition Size
A	Striped Mirror (1+0)	18 GB	6	54 GB

RAID Partition	RAID Level	Type Of Disk	Number Of Disks	Partition Size

[See full-sized image.](#)

SQL Server 2000 Failover Cluster Disk Configuration Worksheet

This worksheet will serve as an easy reference when configuring the operating system. The operating system should not be placed on the shared cluster disk. There needs to be at least two logical disks configured: one for the quorum, and one for SQL Server data (at least one per instance). Anything else is dependant upon the specific needs of your configuration. Also remember to take any mapped/shared drives, CD-ROMs, etc. that would need to be configured into account when filling out this sheet. All disks must be configured as basic, not dynamic, and must be formatted with NTFS. Use this in conjunction with the "Shared Cluster Disk Partition Configuration Worksheet."

Logical Disk is the drive letter that the operating system will use, and that SQL Server will recognize. **Size** is the size of the logical disk. **RAID Partition** is the RAID Partition to use (see the "Shared Cluster Disk Partition Configuration Worksheet") – you can put more than one Logical Disk on a RAID Partition. **Owned By** is for entering the owner of the particular Logical Disk. **Use** is for entering the purpose of the drive.

Example entries:

Logical Disk	Size	RAID Partition	Owned By	Use
C:\	16 GB	A	Local system	OS, page file, SQL Server executables - internal drives
Q:\	500 MB	B	Server Cluster	Quorum - size of physical partition does not matter
R:\	54 GB	C	Instance 1	Transaction Log
S:\	162 GB	D	Instance 1	Data Files

[See full-sized image.](#)

This example shows; for example, drive S:\ which will be formatted as a 162 GB data partition. Initially, there is no need to create a device for your SQL Server 2000 database to fill up the entire 162 GB if it is not necessary. Size your devices appropriately and plan for growth.

Logical Disk	Size	RAID Partition	Owned By	Use
C:\				
D:\				
E:\				
F:\				
G:\				
H:\				
I:\				
J:\				
K:\				
L:\				
M:\				
N:\				
O:\				
P:\				
Q:\				
R:\				
S:\				
T:\				
U:\				
V:\				
W:\				
X:\				
Y:\				
Z:\				

[See full-sized image.](#)

[↑ Top of page](#)

Appendix D – Pre- and Post-Installation Checklists

Pre-Installation Windows Clustering Installation Checklist

This checklist will assist you in verifying you are ready to install your server cluster.

Activity	Pass	Fail
Ping the local loopback address 127.0.0.1	<input type="checkbox"/> Node 1 <input type="checkbox"/> Node 2 <input type="checkbox"/> Node 3 (Datacenter only) <input type="checkbox"/> Node 4 (Datacenter only)	<input type="checkbox"/> Node 1 <input type="checkbox"/> Node 2 <input type="checkbox"/> Node 3 (Datacenter only) <input type="checkbox"/> Node 4 (Datacenter only)
Ping the individual address of each network adapter on each node from each node	<input type="checkbox"/> Node 1 <input type="checkbox"/> Node 2 <input type="checkbox"/> Node 3 (Datacenter only) <input type="checkbox"/> Node 4 (Datacenter only)	<input type="checkbox"/> Node 1 <input type="checkbox"/> Node 2 <input type="checkbox"/> Node 3 (Datacenter only) <input type="checkbox"/> Node 4 (Datacenter only)
Ping the individual address of each network adapter on each node from a machine not in the cluster	<input type="checkbox"/>	<input type="checkbox"/>
Ping the name of each individual node from each node of the cluster	<input type="checkbox"/> Node 1 <input type="checkbox"/> Node 2 <input type="checkbox"/> Node 3 (Datacenter only) <input type="checkbox"/> Node 4 (Datacenter only)	<input type="checkbox"/> Node 1 <input type="checkbox"/> Node 2 <input type="checkbox"/> Node 3 (Datacenter only) <input type="checkbox"/> Node 4 (Datacenter only)
Ping the name of each individual node from a machine not in the cluster	<input type="checkbox"/>	<input type="checkbox"/>
Verify that the RPC service is running on all nodes	<input type="checkbox"/> Node 1 <input type="checkbox"/> Node 2 <input type="checkbox"/> Node 3 (Datacenter only) <input type="checkbox"/> Node 4 (Datacenter only)	<input type="checkbox"/> Node 1 <input type="checkbox"/> Node 2 <input type="checkbox"/> Node 3 (Datacenter only) <input type="checkbox"/> Node 4 (Datacenter only)
Verify cluster disks can be seen by all nodes (power each server on one at a time)	<input type="checkbox"/> Node 1 <input type="checkbox"/> Node 2 <input type="checkbox"/> Node 3 (Datacenter only) <input type="checkbox"/> Node 4 (Datacenter only)	<input type="checkbox"/> Node 1 <input type="checkbox"/> Node 2 <input type="checkbox"/> Node 3 (Datacenter only) <input type="checkbox"/> Node 4 (Datacenter only)
Verify the cluster disks are formatted with NTFS	<input type="checkbox"/>	<input type="checkbox"/>
Domain-level administrator account created for administering the server cluster	<input type="checkbox"/>	<input type="checkbox"/>
WINS server configured	<input type="checkbox"/>	<input type="checkbox"/>
DNS server configured	<input type="checkbox"/>	<input type="checkbox"/>
/3GB or /PAE settings added to boot.ini (if using extended memory)	<input type="checkbox"/>	<input type="checkbox"/>
NetBIOS disabled for Heartbeat NICs	<input type="checkbox"/> Node 1 <input type="checkbox"/> Node 2 <input type="checkbox"/> Node 3 (Datacenter only) <input type="checkbox"/> Node 4 (Datacenter only)	<input type="checkbox"/> Node 1 <input type="checkbox"/> Node 2 <input type="checkbox"/> Node 3 (Datacenter only) <input type="checkbox"/> Node 4 (Datacenter only)
NetBIOS enabled for Public NICs	<input type="checkbox"/> Node 1 <input type="checkbox"/> Node 2 <input type="checkbox"/> Node 3 (Datacenter only) <input type="checkbox"/> Node 4 (Datacenter only)	<input type="checkbox"/> Node 1 <input type="checkbox"/> Node 2 <input type="checkbox"/> Node 3 (Datacenter only) <input type="checkbox"/> Node 4 (Datacenter only)
Network card settings configured (i.e. speed, duplex, etc.)	<input type="checkbox"/> Node 1 <input type="checkbox"/> Node 2 <input type="checkbox"/> Node 3 (Datacenter only) <input type="checkbox"/> Node 4 (Datacenter only)	<input type="checkbox"/> Node 1 <input type="checkbox"/> Node 2 <input type="checkbox"/> Node 3 (Datacenter only) <input type="checkbox"/> Node 4 (Datacenter only)

[See full-sized image.](#)

Post-Installation Windows Clustering Installation Checklist

This checklist will assist you in verifying your server cluster.

Activity	Pass	Fail
Ping the Cluster IP address from all nodes in the cluster	<input type="checkbox"/> Node 1 <input type="checkbox"/> Node 2 <input type="checkbox"/> Node 3 (Datacenter only) <input type="checkbox"/> Node 4 (Datacenter only)	<input type="checkbox"/> Node 1 <input type="checkbox"/> Node 2 <input type="checkbox"/> Node 3 (Datacenter only) <input type="checkbox"/> Node 4 (Datacenter only)
Ping the Cluster IP address from a machine not in the cluster	<input type="checkbox"/>	<input type="checkbox"/>
Ping the Cluster Network Name from all nodes in the cluster	<input type="checkbox"/> Node 1 <input type="checkbox"/> Node 2 <input type="checkbox"/> Node 3 (Datacenter only) <input type="checkbox"/> Node 4 (Datacenter only)	<input type="checkbox"/> Node 1 <input type="checkbox"/> Node 2 <input type="checkbox"/> Node 3 (Datacenter only) <input type="checkbox"/> Node 4 (Datacenter only)
Ping the Cluster Network Name from a machine not in the cluster	<input type="checkbox"/>	<input type="checkbox"/>
Ping all Heartbeat IP address from all cluster nodes	<input type="checkbox"/> Node 1 <input type="checkbox"/> Node 2 <input type="checkbox"/> Node 3 (Datacenter only) <input type="checkbox"/> Node 4 (Datacenter only)	<input type="checkbox"/> Node 1 <input type="checkbox"/> Node 2 <input type="checkbox"/> Node 3 (Datacenter only) <input type="checkbox"/> Node 4 (Datacenter only)
MS DTC configured (if Windows 2000, comclust run on each node)	<input type="checkbox"/> Node 1 <input type="checkbox"/> Node 2 <input type="checkbox"/> Node 3 (Datacenter only) <input type="checkbox"/> Node 4 (Datacenter only)	<input type="checkbox"/> Node 1 <input type="checkbox"/> Node 2 <input type="checkbox"/> Node 3 (Datacenter only) <input type="checkbox"/> Node 4 (Datacenter only)
Fail MS DTC to other nodes	<input type="checkbox"/> Node 1	<input type="checkbox"/> Node 1
	<input type="checkbox"/> Node 2 <input type="checkbox"/> Node 3 (Datacenter only) <input type="checkbox"/> Node 4 (Datacenter only)	<input type="checkbox"/> Node 2 <input type="checkbox"/> Node 3 (Datacenter only) <input type="checkbox"/> Node 4 (Datacenter only)
Fail cluster disk resources to other nodes	<input type="checkbox"/> Node 1 <input type="checkbox"/> Node 2 <input type="checkbox"/> Node 3 (Datacenter only) <input type="checkbox"/> Node 4 (Datacenter only)	<input type="checkbox"/> Node 1 <input type="checkbox"/> Node 2 <input type="checkbox"/> Node 3 (Datacenter only) <input type="checkbox"/> Node 4 (Datacenter only)

[See full-sized image.](#)

Pre-SQL Server 2000 Virtual Server Installation Checklist

This checklist will assist you in verifying you are ready to install your SQL Server 2000 virtual server.

Activity	Pass	Fail
No Errors That May Interfere With Install in Event Viewer	<input type="checkbox"/>	<input type="checkbox"/>
Domain account(s) created for SQL Server administration (not necessary to be domain admin)	<input type="checkbox"/>	<input type="checkbox"/>
All resources moved to node which is going to initiate the installation process of SQL Server	<input type="checkbox"/>	<input type="checkbox"/>
All nodes have access to the installation path (if using a network share)	<input type="checkbox"/>	<input type="checkbox"/>

[See full-sized image.](#)

Post-SQL Server 2000 Virtual Server Installation Checklist

This checklist will assist you in verifying your SQL Server 2000 virtual server installation.

Activity	Pass	Fail
All SQL Server resources online and configured	<input type="checkbox"/>	<input type="checkbox"/>
All dependencies configured properly	<input type="checkbox"/>	<input type="checkbox"/>
Ping SQL Server virtual server IP address(es) from all nodes in the cluster	<input type="checkbox"/> Node 1 <input type="checkbox"/> Node 2 <input type="checkbox"/> Node 3 (Datacenter only) <input type="checkbox"/> Node 4 (Datacenter only)	<input type="checkbox"/> Node 1 <input type="checkbox"/> Node 2 <input type="checkbox"/> Node 3 (Datacenter only) <input type="checkbox"/> Node 4 (Datacenter only)
Ping SQL Server virtual server IP address(es) from a machine not in the cluster	<input type="checkbox"/>	<input type="checkbox"/>
Ping SQL Server virtual server network name from all nodes in the cluster	<input type="checkbox"/> Node 1 <input type="checkbox"/> Node 2 <input type="checkbox"/> Node 3 (Datacenter only) <input type="checkbox"/> Node 4 (Datacenter only)	<input type="checkbox"/> Node 1 <input type="checkbox"/> Node 2 <input type="checkbox"/> Node 3 (Datacenter only) <input type="checkbox"/> Node 4 (Datacenter only)
Ping SQL Server virtual server IP address(es) from a machine not in the cluster	<input type="checkbox"/>	<input type="checkbox"/>
Fail resources for the SQL Server virtual server to all nodes of the VS definition	<input type="checkbox"/> Node 1 <input type="checkbox"/> Node 2 <input type="checkbox"/> Node 3 (Datacenter only) <input type="checkbox"/> Node 4 (Datacenter only)	<input type="checkbox"/> Node 1 <input type="checkbox"/> Node 2 <input type="checkbox"/> Node 3 (Datacenter only) <input type="checkbox"/> Node 4 (Datacenter only)
Run select * from ::fn_virtualservicenodes() query	<input type="checkbox"/>	<input type="checkbox"/>

[See full-sized image.](#)

[↑ Top of page](#)

[Manage Your Profile](#)

© 2008 Microsoft Corporation. All rights reserved. [Contact Us](#) | [Terms of Use](#) | [Trademarks](#) | [Privacy Statement](#)

Microsoft