### IP Address

SQL Server Configuration Manager will display the properties of the TCP/IP protocol.

and, it does not display the clustered IP address or addresses that are configured during the installation process. SQL Server Configuration Manager will display all of the other IP addresses bound to the network card that the SQL Server 2005 failover clustering instance is bound to.

To view the IP address for your clustered installations of SQL Server 2005 or Analysis Services, we have to perform the following steps:

1. Start Cluster Administrator.

In the left pane,

Expand **Groups**

and select the cluster group that contains the desired SQL Server 2005 instance.

In the right pane, double-click the appropriate IP resource, which will be named

SQL IP Address *n* (*sqlname*), where *n* is the number of the IP address, and *sqlname* is the name of the failover clustering instance you configured during installation.

1. Click the **Parameters** tab to view the IP address properties.

### Node Configuration and Information

SQL Server 2005 introduces the DMV sys.dm\_os\_cluster\_nodes, which has the same functionality as the function fn\_virtualservernodes in SQL Server 2000. This DMV returns all of the server cluster nodes that are defined as part of the failover clustering instance, and can be used to verify the configuration. The user must have the VIEW SERVER STATE permission to access this DMV.

To execute, run SELECT \* FROM sys.dm\_os\_cluster\_nodes in a query window. For more information, see [sys.dm\_os\_cluster\_nodes](http://go.microsoft.com/fwlink/?LinkId=71426).

Note: fn\_virtualservernodes is still available and supported in SQL Server 2005, but Microsoft recommends changing code that references it to use the new DMV. To access fn\_virtualservernodes in SQL Server 2005, the user must have the VIEW SERVER STATE permission.

You can also view which physical node currently owns the resources by executing SELECT SERVERPROPERTY('ComputerNamePhysicalNetBIOS') (the quotes are required) in a query window.

## Adding or Removing a Node

There are cases where you may need to remove, a node from the cluster.

There are two levels of removal when it comes to SQL Server: removing it from the of the SQL Server failover clustering instance, and then removing the node from the Windows server cluster itself.

Important: If you must evict the node from the server cluster, do not attempt to remove the node via Cluster Administrator before running the SQL Server-specific steps. Doing so will damage your SQL Server installation. You must remove the node from the SQL Server definition before you evict the node in Cluster Administrator.

Similarly, if you need to add a node into the definition of the failover clustering instance, you would execute similar steps.

### Using Setup

The process for adding or removing a node is similar to setting up a new clustered instance of SQL Server. In Control Panel, open **Add or Remove Programs**, select **Microsoft SQL Server 2005**, and click **Change**. Setup launches; at the Change or Remove Instance page, select **Maintain the Virtual Server**, and then follow the instructions in Step 14 of Appendix A to add or remove a node.

## Changing the IP Address of SQL Server

SQL Server 2000 required re-running Setup to change the IP address. SQL Server 2005 now uses Cluster Administrator. For instructions, see [How to: Change the IP Address of a SQL Server 2005 Failover Cluster](http://go.microsoft.com/fwlink/?LinkId=71423).

Note: Changing the IP address will require a small amount of downtime and the proper privileges at the Windows level.

## Uninstalling a Clustered Instance

Removing a clustered instance of SQL Server 2005 is different from SQL Server 2005. There are two options for uninstalling a SQL Server 2005 failover clustering instance.

* Remove it via the command prompt as described in [How to: Install SQL Server 2005 from the Command Prompt](http://go.microsoft.com/fwlink/?LinkID=71413).
* Remove it via Setup as detailed in [How to: Remove a SQL Server 2005 Failover Clustered Instance (Setup)](http://go.microsoft.com/fwlink/?LinkId=71422).

If there are problems in the uninstall process, see [How to: Manually Uninstall a SQL Server 2005 Failover Cluster](http://go.microsoft.com/fwlink/?LinkId=71421).

Removing the instance itself does not remove the client tools and utilities.

To uninstall a clustered instance in SQL Server 2005, you must meet these requirements:

* You must be logged in as the cluster administrator account or another account with the proper administrative privileges when uninstalling SQL Server to allow the Setup process to execute with the right permissions.
* The SQL Server installation CD or installation folder must be accessible.
* The process must be initiated from the node that currently owns the instance’s resources.

Important: If you uninstall SQL Server 2005, it is recommended that you do not remove SQL Native Client if you had configured a side-by-side installation that contains both SQL Server 2000 and SQL Server 2005 instances in the same cluster. If you remove the SQL Native Client, you will render your SQL Server 2000 instance useless since the SQL Server 2000 instance now depends on the SQL Native Client. You will need to reinstall SQL Native Client to get any other instance of SQL Server working again.

## Applying a SQL Server Service Pack

Before installing a service pack on a clustered installation of SQL Server or Analysis Services, consult the readme that is associated with that particular service pack. It may contain specific information and steps that may need to be addressed outside of the basic installation steps.

A SQL Server 2005 service pack is applied on a per-instance basis, which means you may need to run the installer more than once if the instance's resources are owned by another node. Updates to Books Online may be packaged separately from the service pack itself.

**Important** Since the management tools may only be installed on the node which initiated the original installation, if the management tools are installed on other nodes, those must be patched separately from the service pack install on the instance itself.

For instructions on installing a SQL Server 2005 service pack in a clustered environment via the graphical user interface (GUI), see Appendix D. Service pack installs can also be automated. For instructions, consult the readme that ships with the service pack.

### SQL Server Configuration Manager

SQL Server 2005 provides a new tool, SQL Server Configuration Manager, that replaces the earlier SQL Server Service Manager. SQL Server 2005 SQL Server Configuration Manager is fully cluster-aware and it can be used to stop and start all SQL-related services. SQL Server Configuration Manager is the recommended way to stop and start SQL Server.

1. To stop and start a SQL Server 2005 failover clustering instance using SQL Server Configuration Manager, follow these steps:
2. Start SQL Server Configuration Manager.
3. Expand **SQL Server 2005 Services** in the left pane.
4. Right-click the resource whose status you want to change, and select **Start**, **Stop**, **Pause**, or **Restart**.

Note: SQL Server Configuration Manager does not refresh itself very frequently, so it may not reflect the current status. You will have to manually refresh the status.

## Starting, Stopping, and Pausing Services

Although there are several tools for starting and stopping SQL Server 2005 in a cluster, SQL Server Management Studio is not one of them. The three tools for starting and stopping SQL Server 2005 in a cluster are:

* SQL Server Configuration Manager
* SQL Server Surface Area Configuration
* Cluster Administrator

Important: As with previous versions of SQL Server failover clustering, services such as SQL Server Agent depend on SQL Server to start. If you manually stop the SQL Server service, it will stop any dependencies, but upon restart, you will manually have to restart them.

## Connecting to a SQL Server 2005 Failover Cluster

Use the *<FAILOVER\_CLUSTERING\_INSTANCE\_NAME>\<INSTANCE NAME>* or the IP address to connect to a clustered instance of SQL Server running on a failover cluster. There is no change from previous behavior. As with previous versions of failover clustering, it is not possible to access a clustered instance of SQL Server by using the underlying node name that the instance is currently running on. SQL Server 2005 will not listen on the IP address of the local server; it will listen only on the clustered IP addresses created during the installation of a clustered instance of SQL Server. If the virtual IP address does not have DNS registration, SQL Server will not listen on that IP address.

For a cluster, the resources are enumerated, the network name that is needed is found (as well as the IP addresses on which the network name depends), and then the SQL Server instance listens on those IP addresses. This change in SQL Server 2005 keeps all the IP modifications external and worry-free; in a case where customers must add an IP for a remote connection on a temporary basis and may be only using a host file, there should be no issues.

Note: By default, TCP/IP and named pipes are enabled in a failover cluster, though they are not disabled by default on a stand- alone SQL Server installation.

### Verify Clustered Resource Failover

After installation is complete, you must properly test it to ensure that it is working as expected.

1. Start Cluster Administrator.
2. Right-click the resource group containing the SQL Server or Analysis Services installation, select **Move Group**, and then select another node of the cluster. SQL Server should now stop on the node it was owned by and start on the other node.
3. Repeat step 2 for each node of the cluster.
4. Ping the SQL Server and/or Analysis Services IP address from all nodes within the cluster as well as from a computer outside of the cluster to ensure IP address resolution.
5. Ping the SQL Server and/or Analysis Services name from all nodes within the cluster as well as from a computer outside of the cluster to ensure name resolution.

## Troubleshooting Installation Problems

If installation fails, check Windows Event Logs, and also the logs created during the installation process.

The logs can be found in the SQL Server program directory under the \90\Setup Bootstrap\LOG subdirectory.

SQL Server will be installed to C:\Program Files\Microsoft SQL Server. Under \LOG, there will be one or more numbered .cab files. Under the Files directory are the uncompressed log files.

The .cab file contains all of the logs from the installation, and has a naming convention of SqlSetup*nnnn*.cab, where *nnnn* is a unique number.

The highest numbered .cab file will be the latest installation.

The directory will also contain a file named Summary.txt, which corresponds to the latest installation and will contain information about any warnings detected during the Setup process.

For more troubleshooting tips, see [Failover Cluster Troubleshooting](http://go.microsoft.com/fwlink/?LinkId=71513).

General troubleshooting steps where the cause of failure is not clear include:

* Flushing the name resolution cache. At a command prompt, enter the following two commands:  
    
  Ipconfig /flushdns   
  Nbtstat –RR
* Verifying that 8.3 naming structure is enabled.
* Open a command-prompt window and navigate to your profile directory. At the command prompt, enter DIR /X. Both long and 8.3 names should be returned. If no 8.3 naming is returned, you will need to edit the following registry key:  
    
  HKLM\SYSTEM\CURRENTCONTROLSET\CONTROL\FileSystem\NtfsDisable8dot3NameCreation  
    
  The NtfsDisable8dot3NameCreation value must be set to 0; if the value is changed, the computer will need to be rebooted and any directories previously created will need to be removed before running Setup. If these directories are associated with installed components, those components must be uninstalled before removing the directories.
* If the instance of SQL Server 2005 being installed uses the same name as a computer or SQL Server instance already registered in with Active Directory and/or DNS, the name must manually be purged of that name before running Setup. This task will need to be done by a qualified administrator.

## How Failover Clustering Works

The clustered nodes use a "heartbeat" signal to check whether each node is alive, at both the operating system level and the SQL Server level. At the operating system level, the nodes in the cluster are in constant communication, validating the health of all the nodes.

After installing a SQL Server failover cluster, the node hosting the SQL Server resource uses the Service Control Manager to check every 5 seconds whether the SQL Server service appears to be running.

This "LooksAlive" check does not impact the performance of the system, but also does not do a thorough check; the check will succeed if the service appears to be running even though it might not be operational. Because the LooksAlive check does not do a thorough check, a deeper check must be done periodically; this "IsAlive" check runs every 60 seconds.

The IsAlive check runs a SELECT @@SERVERNAME Transact-SQL query against SQL Server to determine whether the server can respond to requests. Although a reply to the IsAlive query confirms that the SQL Server service is available for requests, it does not guarantee that all user databases are available, or that the user databases are operating within necessary performance/response-time requirements.

If the IsAlive query fails, the IsAlive health check is retried five times and then it 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, the failover cluster will attempt to either restart the resource on the same node or it will fail over to another available node. The IsAlive query tolerates a few errors, but ultimately it fails if its threshold is exceeded.

During failover of the SQL Server instance, SQL Server resources start up on the new node. Windows clustering starts the SQL Server service for that instance on the new node and SQL Server goes through the recovery process to start the databases. 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 (the Redo phase), and any incomplete transactions are rolled back (the Undo phase). In SQL Server 2005 Enterprise Edition, each user database will be available to the user once the Redo phase completes; for the other editions, as with all previous versions, each user database is unavailable until the Undo phase completes. The length of the recovery process depends on how much activity must be rolled forward or rolled back upon startup. The 'recovery interval' sp\_configure option of the server can be set to a low number to avoid longer Redo recovery times and to speed up the failover process. The Undo recovery time can be reduced by using shorter transactions so that any uncommitted transactions do not have much to roll back.

# Failover Clustering Basics

SQL Server 2005 failover clustering is built upon an already established Windows failover cluster. There are three main types of clusters in a Windows environment:

* Failover cluster

A failover cluster can consist of up to as many as eight 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, to provide high availability, scalability, and manageability for resources and applications. When a server, resource, or cluster-aware application within the cluster becomes unavailable, the resources associated with that clustered application (such as SQL Server) are transferred to a server that is available.

A failover cluster comprises one or more nodes, or servers, with a set of shared cluster disks specifically configured for use with the cluster. Applications such as SQL Server are coded to be cluster-aware as a resource DLL, and installed into Windows failover cluster group, known as a resource group. The resource group contains the specific related cluster resources for a given application. At any given time, each resource group and its resources can only be owned by a single node in the cluster. The application service has a failover clustered instance name that is independent of the node names, and for that reason is referred to as a "failover clustered instance." An application can connect to the failover instance by referencing the instance name, without having to know which node hosts the failover instance.

Windows Server 2003 has two types of cluster quorum models: the traditional shared quorum failover cluster, which uses a shared disk to hold the cluster state, and the local quorum or Majority Node Set (MNS) cluster, which uses a local disk on each node to hold the cluster state. SQL Server 2005 supports both cluster quorum models.

Important: Before the release of SQL Server 2005, a failover cluster was only a SQL Server-specific term. At the Windows level, a cluster for availability was known as a server cluster (or MSCS). Going forward, availability clustering at the Windows level will also be known as a failover cluster.

* 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 configuration, 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 distributes the load to another server. SQL Server 2005 failover clustering is not a load balanced solution, and is not installed or configured using Network Load Balancing. A SQL Server 2005 failover clustering installation can be part of an overall architecture in which a Web farm using a Network Load Balancing cluster connects to a failover cluster. If 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. SQL Server failover clustering is not built on top of a Network Load Balancing cluster, and this type of cluster cannot be built on the same hardware as a server cluster.

* Windows Compute Cluster Server 2003

Windows Compute Cluster Server 2003 is a new edition of Windows which is optimized for high performance computing. SQL Server does not currently work with this edition of Windows, and it is fully unsupported to configure Windows Compute Cluster Server 2003 on the same hardware as failover clustering.

The failover clustering instance is the same as an SQL Server instance. The previous terminology was known as a "virtual server," but the has been changed to avoid confusion with the Microsoft Virtual Server product.

As noted above, a Windows failover cluster has one or more nodes configured with specific resources in resource groups. When a SQL Server failover clustering instance is installed on a Windows failover cluster, all of its resources are contained within one of those groups on the cluster that is not shared with anything else, such as the quorum or Microsoft Distributed Transaction Coordinator (MS DTC). From a strict definition, a SQL Server failover clustering instance consists of these resources:

* A combination of one or more cluster disks.
* A network name (the failover clustering instance name).
* One or more IP address cluster resources.

The name of the SQL Server is defined by the network name cluster resource, and that name will be used by applications and end users to connect to the failover instance; together these resources represent a logical Windows Server on the network, while running across one or more real Windows Server computers. A failover instance of SQL Server is an item that is created during SQL Server setup; it is not provided by Windows Server.

A given cluster can contain many instances of SQL Server, each in its own cluster group. Each cluster group can contain at most one instance of SQL Server. The SQL Server failover instance will then be installed into the cluster group selected during SQL Server setup.

If the administrator uninstalls the instance of SQL Server 2005 from within a failover instance, the failover instance, including all IP addresses and the network name, is removed from the failover cluster group as well.