CHAPTER 18

Administering and Managing Failover Clustering

Failover clustering is one of four SQL Server 2005 high-availability alternatives. Other SQL Server 2005 high-availability alternatives include database mirroring, log shipping, and peer-to-peer replication. Database mirroring and peer-to-peer replication are new technologies introduced with SQL Server 2005 and are discussed in other chapters within this book.

This chapter first focuses on an overview of SQL Server clustering, Windows clustering prerequisites, and ways to configure Windows clustering with Windows Server 2003 R2. The chapter also includes step-by-step procedures on installing and configuring both a single instance and multiple instances of SQL Server 2005 two-node failover clusters for the Database Engine and Analysis Services components. The final portion of the chapter includes administrative tasks on how to manage a SQL Server failover cluster.

SQL Server 2005 Failover Clustering Overview

Windows Server 2003 and SQL Server 2005 both support the *shared-nothing cluster model*. In a shared-nothing cluster, two or more independent servers share resources; however, each computer owns and manages its local resources and provides nonsharing data services. In case of a node failure, the disks, resources, and services running on the failed node fail over to a surviving node in the cluster. The scenario is a little different with SQL Server high-availability clustering. Specifically, only one node manages one particular set of disks and services at any given time.

Note

SQL Server failover clustering provides high availability for mission-critical databases and server nodes. However, it does not replace the need for a comprehensive backup and recovery strategy for a SQL Server environment. Equally important, you should not rely on clustering as a replacement for disaster recovery and a solid backup and recovery strategy.

Determining When to Implement a SQL Server 2005 Failover Cluster

Typically, organizations implement a SQL Server 2005 failover cluster to address the following situations:

- To increase server availability for mission-critical applications and network services
- To provide high-availability support for an entire instance of SQL Server and not just a database
- To provide a seamless failover that does not affect client applications and end users
- To provide an automatic failover that does not require database administrator intervention
- To reduce downtime during routine maintenance or unplanned failures
- To perform service pack rolling upgrades

Failover Clustering Terminology

Before installing SQL Server 2005 failover clustering, you need to understand that Windows and SQL clustering come with different levels of availability or options. Determine which fault-tolerant option is best by using the following list, which describes some of the options, describes how they work, and explains a few traditional terms relevant to the way SQL Server 2005 clustering operates:

- **SQL Server Virtual Server**—A SQL Server virtual server is, in fact, a cluster-configured resource group that contains all resources necessary for SQL Server to operate on the cluster. This includes the NetBIOS name of the virtual server, a TCP/IP address for the virtual server and all disk drives, and vital SQL Server services required to operate in a clustered configuration. In a multiple instance, two or more node clusters and one SQL Server virtual server are created per node, while the

NetBIOS name and TCP/IP address of the cluster form the virtual server. When failover occurs in this configuration, the entire SQL Server virtual server fails over to the surviving node in the cluster dynamically.

- **Heartbeat**—A single User Datagram Protocol (UDP) packet is sent every 500 milliseconds between nodes in the cluster across the internal private network. This packet relays health information about the cluster nodes as well as health information about the clustered application. If there is no response during a heartbeat to indicate the node is alive, the cluster begins the failover process. In SQL Server 2005, this interval can be changed. This capability is useful when you are using a geographically dispersed cluster.
- **Failover**—Failover is the process of one node in the cluster changing states from offline to online, resulting in the node taking over responsibility of the SQL Server virtual server. The Cluster Service fails over a group in the event that node becomes unavailable or one of the resources in the group fails.
- **Failback**—Failback is the process of moving back a SQL Server virtual server that failed over in the cluster to the original online node.
- **Quorum Resource**—The quorum resource is the shared disk that holds the cluster server's configuration information. All servers must be able to contact the quorum resource to become part of a SQL Server 2005 cluster.
- **Resource Group**—A resource group is a collection of cluster resources such as the SQL Server NetBIOS name, TCP/IP address, and the services belonging to the Exchange cluster. A resource group also defines the items that fail over to the surviving nodes during failover. These items also include cluster resource items such as a cluster disk. It is also worth noting that a resource group is owned by only one node in the cluster at a time.
- **Cluster Resource**—Cluster resources contain vital information for the SQL Server virtual server and include its network TCP/IP addresses, NetBIOS name, disks, and SQL Server services, such as the System Attendant. These cluster resources are added to cluster groups when the virtual server is created to form SQL Server virtual servers.
- **Dependency**—A dependency is specified when creating cluster resources. Similar to a dependency on SQL Server services, a cluster resource identified as a dependency indicates a mandatory relationship

exists between resources. Before a cluster resource is brought online, the resource defined as a dependent must be brought online first. For instance, the virtual server NetBIOS name is dependent on the TCP/IP address; therefore, the TCP/IP address of the virtual server must be brought online before the NetBIOS name is brought online.

- **Majority Node Cluster**—In this configuration, each node is responsible for contributing one local disk to the quorum disk set that is used as storage disks. This configuration limits the majority-of-node-resource to one owner at a time. Because the quorum does not require a shared disk, the solution is typically used for geographically dispersed clusters.
- **Cluster Administrator**—The Cluster Administrator is a tool used by cluster and database administrators for accessing, creating, and administering Windows clusters. The Cluster Administrator is included by default with Windows Server 2003 and can be launched from any active node within the cluster. Additional administration and management tasks include viewing, creating, and deleting cluster groups, cluster resources, and nodes.
- **Quorum-Device Cluster**—Using the quorum type resource requires the cluster storage resource to be connected with a fiber channel or small computer system interface (SCSI) bus. In this configuration, any physical disk can be configured as a quorum disk.

SQL Server Failover Clustering Prerequisites

SQL Server failover clustering is based on the Windows shared-nothing model. Two or more nodes can control a single set of media that hold the application data. In the case of SQL Server 2005, this refers to a virtual instance where there is only a single copy of the database and logs residing on shared disks. Multiple nodes are available and can control these resources one at a time.

Before installing SQL Server clustering, ensure the following prerequisites are met:

- A minimum of two identical servers running Windows Server 2003 Enterprise Edition is required.
- Two network interfaces are needed per server: one for the private heartbeat and the other for the public network.

- Shared disk storage is needed for the quorum, database, and log files. This could be a storage area network (SAN), small computer system interface (SCSI), or Internet SCSI (ISCSI) storage solution. All shared disks must be configured as basic because clustering does not support dynamic disks. In addition, all shared disks must be online, configured with NTFS, and be seen from all nodes.
- Microsoft Distributed Transaction Coordinator (MSDTC) must be installed and configured prior to the installation of SQL Server. MSDTC is required if the following components are installed: Database Engine and Integration Services, Notification Services, or Workstation components.
- Microsoft Cluster Server (MSCS) must be configured prior to the installation of SQL Server. The clustering groups and resources should be available, operational, and online.
- Separate service accounts are needed for both the Microsoft Cluster Server and SQL Server services.
- All nodes should be identical and have the same service pack hotfixes and identical software.
- NETBIOS must be disabled on all network adapters being used for the private heartbeat network.
- Domain groups should be created for the clustered services to be used based on the SQL Server components that will be installed as part of the cluster.
- All hardware and software being used must be certified by Microsoft and be on its Windows Catalog and Hardware Compatibility List (HCL).

Failover Clustering Alternatives

The following bullets depict the types of clustering alternatives available with SQL Server 2005:

- **Single-Instance Failover**—In a SQL Server 2005 single-instance failover configuration, the cluster runs a single virtual instance of SQL Server on all nodes in the cluster. Only one node in the cluster can service end users at any one time; this is known as the *active node*. The passive node, on the other hand, is on standby. If a failure occurs, the clustered resources are shifted to the passive node, which then begins to service clients. In this configuration, one virtual SQL Server instance is configured and shared by one or both nodes.

- **Multiple-Instance Failover**—In a multiple-instance failover configuration, also known as *active/active clustering*, each active node runs its own virtual instance of SQL Server. Each instance of SQL Server includes a separate installation of the full service and can be managed, upgraded, and stopped independently. If you want to apply a multiple-instance failover configuration, at least two instances of SQL Server need to be installed on the cluster and each instance should be configured to run on a certain node as its primary server.
- **N+1**—This is a deviation of the multiple-instance failover clustering topology just discussed. In this scenario, more than two nodes are configured within the failover cluster solution and share the same failover node in the event of a failure. For example, in a four-node cluster, there may be three active nodes and one passive node. The passive node acts as a hot standby for any or all of the three active nodes. This solution reduces hardware costs because there isn't a one-to-one mapping between active and passive nodes. However, the major disadvantage is that the passive node must have enough hardware resources to take on the load of all three active nodes if they crash at the same time. The chances of three nodes crashing is highly unlikely; however, in the computer industry, we all know there is a first for everything.
- **N+M**—Sometimes there is more than one active node in the cluster, so having a single dedicated failover node such as in the N+1 scenario is not sufficient enough to provide redundancy. Therefore, more than one standby node (M) is included in the cluster and available for failover. The number of standby servers is a trade-off between cost and reliability requirements.
- **Geographically Dispersed Clusters**—SQL Server 2005 also offers geographically dispersed clusters. This scenario does not require a quorum drive to be configured on a shared disk, thus allowing active and passive nodes to be in separate physical locations. If you want to implement this solution, specific hardware is required from a vendor. This hardware must be certified from the vendor and Microsoft. This is a different list from the Microsoft Clustering HCL. Implementing geographically dispersed clusters is expensive. It is recommended to use database mirroring instead of building a geographical cluster because database mirroring is much cheaper and also provides high availability, seamless failover, and automatic client failover.

Note

Server load and performance degradation should be analyzed when working with multiple instances within a single SQL Server failover cluster. You must ensure that the surviving node can handle the load if running more than one SQL Server instance on a single server.

SQL Server 2005 Failover Clustering Scalability Metrics

A discussion on cluster basics is always essential because it can help organizations define a suitable operating system for their business. SQL Server 2005 Enterprise Edition on Windows 2003 Enterprise or Windows 2003 Data Center Edition can support up to eight nodes within a single cluster. SQL Server 2005 Standard Edition can support up to two nodes. Failover clustering of SQL Server 2005 can be configured in two ways: a single-instance failover (active/passive) configuration or a multiple-instance failover (active/active) configuration. When using multiple-instance failover clustering, SQL Server 2005 scales up to 50 instances on Enterprise Edition, whereas the Standard Edition supports up to 16.

Note

When implementing SQL Server multiple-instance failover clustering, you should be aware that each instance requires a separate virtual server and each virtual server requires a separate cluster group with dedicated resources.

SQL Server 2005 Cluster-Aware Components

The SQL Server 2005 Database Engine, Analysis Services, and Full-Text Search components are cluster aware, which allows them to interact with the server cluster. Unfortunately, Reporting Services, Notification Services, and Integration Services are not cluster aware.

Combining Failover Clustering with Other SQL Server High-Availability Alternatives

Other SQL Server high-availability alternatives can be combined with failover clustering for maximum availability, business continuity, and disaster recovery. For example, CompanyABC may have a two-node failover cluster residing in its Toronto office. This cluster provides high availability for a production database at the server level in that physical location. Regulatory compliances such

as the Sarbanes-Oxley Act may have a requirement that CompanyABC maintain a disaster recovery site, and all production databases must be available in another location in the event of a disaster in the Toronto location. Therefore, CompanyABC can implement database mirroring in conjunction with clustering and mirror the production databases from the Toronto location to its disaster recovery location in San Francisco. The production database would be available in San Francisco in the event of the total cluster failure (which is highly unlikely) or the Toronto site is unavailable.

On a side note, database mirroring is not the only other high-availability alternative that works with failover clustering: Log shipping and replication can also be used.

Administering SQL Server Failover Clustering

The following example illustrates a multiple-instance SQL Server failover installation on a two-node cluster using Windows Server 2003 Enterprise Edition. Two instances of the SQL Server Database Engine and a single instance of Analysis Services are configured in this failover cluster. The two nodes reside in Toronto, the physical site for CompanyABC. Table 18.1 depicts the cluster node, drives, IP addresses, network card, and NetBIOS information used for the remainder of the examples in this chapter.

Table 18.1 **CompanyABC's Clustering Information**

Item	Description
Cluster Nodes	
NODE 1 NETBIOS	TOR-CL01
NODE 2 NETBIOS	TOR-CL02
NODE 1 Public IP Address	192.168.115.204
NODE 2 Public IP Address	192.168.115.203
NODE 1 Private IP Address	10.0.0.1
NODE 2 Private IP Address	10.0.0.2
Cluster Name	TOR-CLUSTER
SQL Server Virtual Instance01	
SQL Server Virtual Instance01 Name	TOR-SQL01\INSTANCE01
SQL Server Virtual IP Address	192.168.115.206
Instance01 Failover Components	Database Engine

Table 18.1 **continued**

Item	Description
SQL Server Virtual Instance02	
SQL Server Virtual Instance02 Name	TOR-SQL02\INSTANCE02
SQL Server Virtual IP Address	192.168.115.207
Instance02 Failover Components	Database Engine
Analysis Services	
Disk Drive/Logical Unit Number (LUNs) Layout on the SAN or Shared Storage	
Quorum	Q Drive
Database Files - Instance01	D Drive
Database Logs - Instance01	L Drive
Database Files - Instance02	E Drive
Database Logs - Instance02	M Drive

Using the information in Table 18.1, you can now turn your attention to preparing the operating system and configuring the Windows Server 2003 failover cluster.

Preparing the Operating System for Each Node

For this example, creating a single-instance or multiple-instance failover cluster starts out with a standard installation of Windows Server 2003, x64 Enterprise Edition. Follow these steps to build the operating system with either Windows Server 2003 Standard or Enterprise Edition:

1. For each node in the two-node cluster, install Windows Server 2003 x64. Use the typical settings when setting up Windows Server 2003.
2. Join the nodes to the domain that will host the SQL Cluster such as CompanyABC.
3. Create a heartbeat network between the two nodes by addressing a network interface card (NIC) on a different IP space from the production network. Isolate these heartbeat NICs to an isolated VLAN or a separate network switch or use a crossover cable. This network will be used by the cluster nodes to communicate with each other. For this example, use the IP addresses based on 10.0.0.1 for node 1 and 10.0.0.2 for node 2. Disable NetBIOS on the heartbeat adapters.
4. Install the latest Windows service packs and hotfixes.

Configuring the Shared Storage for the Failover Cluster

In the case of a two-node multiple-instance failover cluster, you need to configure several shared drives to be accessible by both nodes. These drives include the quorum drive, the MSDTC distributed transaction coordinator drive, 2x database drives and 2x log drives, and the database drive, as shown in Table 18.2.

Table 18.2 **CompanyABC's Clustering Sample Storage Information**

Drive Description	Drive Letter
Quorum	Q Drive
Database Files - Instance01	D Drive
Database Logs - Instance01	L Drive
Database Files - Instance02	E Drive
Database Logs - Instance02	M Drive

> **Note**
>
> This example does not use a dedicated drive for the MSDTC clustered resource. In a production environment, a dedicated shared drive should be allocated.

There are three ways to accomplish shared storage:

- **Network Attached Storage**—Usually connected to the server via Internet SCSI (ISCSI) or virtual logical disk drivers
- **Storage Area Network**—Connected by fiber channel
- **Shared SCSI**—Connected via a separate SCSI controller and attached to a dual-termination enclosure

> **Note**
>
> If using storage area networks to attach the shared resources, be sure to zone the LUNs (virtual disks) so that only the potential owner nodes and disks are present in that zone. This configuration is important because if another Windows system were to see the LUN, it might try to write its own signature to the "disk," which would break the access from the point of view of the cluster nodes.

Creating the Windows Cluster

Now that the nodes are prepared and the shared storage is created, the cluster can be formed. Be sure to have the following items ready for the cluster configuration:

- Unique names for each node
- Unique name for the cluster identity
- Unique name for each SQL Server instance
- Unique IP addresses for each of the names created previously

> **Note**
> All the IPs listed must be from the same network range.

With these items ready, follow these steps to create the cluster:

1. Right-click My Computer and choose Manage.
2. Click the Disk Management icon.
3. When the Initialize and Convert Wizard appears, click Next.
4. Click Next twice to initialize the disks but not convert them, which results in a screen that shows basic disks, similar to the one shown in Figure 18.1.

> **Note**
> Disks made available to a cluster must be configured as basic disks. A Windows 2003 cluster can't access a dynamic disk.

5. Click Finish.
6. Right-click the name of each disk that is currently unpartitioned, create a partition, and establish the size needed. Be sure to create them as primary partitions and format them as NTFS.
7. From the Start menu, click All Programs, Administrative Tools, Cluster Administrator.
8. Under Action, choose Create New Cluster and click OK.
9. When the Setup Wizard launches, click Next.

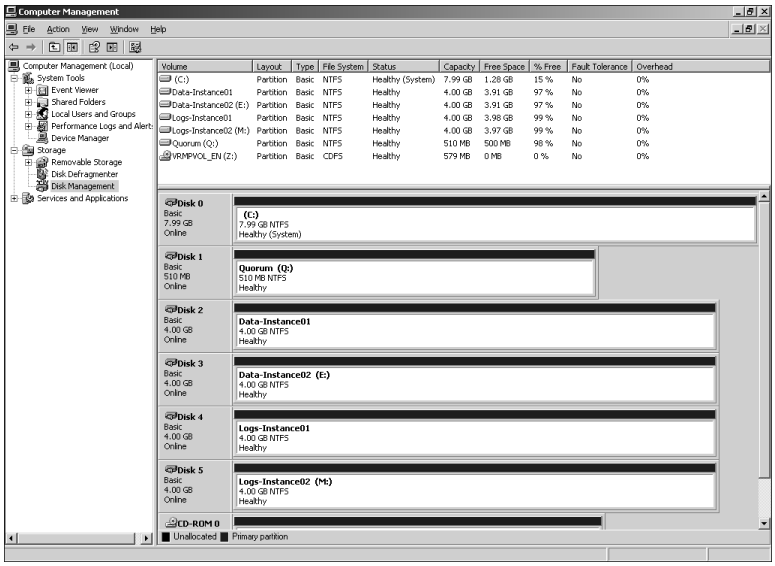

FIGURE 18.1
Shared disks as basic disks.

10. Choose the domain that will host the cluster in the drop-down menu and enter the name of the cluster. This is the name of the cluster itself and doesn't affect the SQL Server Virtual Instance. Click Next.
11. Enter the name of the first node that will participate in the cluster. Click Next.
12. The Setup Wizard analyzes the node to see whether it can become a cluster node. When the analysis is complete, click Next.
13. Enter the IP address that will be used to manage the cluster. This needs to be a unique and reachable IP address. Click Next.
14. Enter the name and password of the account under which the cluster service will run. This account needs to have local administrator rights to each node of the cluster. Click Next.
15. On the Proposed Cluster Configuration screen, click the Quorum button and select the shared disk that was created to host the quorum. Click OK and then click Next.
16. When the Setup Wizard completes its tasks, click Next and then click Finish.

Adding the Second Node to the Windows Cluster

Now that the cluster has been established by the first node, you can join the second node to the cluster. To do this, follow these steps:

1. Right-click Node1 in the left pane of the Cluster Administrator.
2. Click New and then click Node.
3. When the Setup Wizard launches, click Next.
4. Type the name of the second node, click Add and then click Next.
5. The analysis runs to see whether the node selected is a viable cluster node. When the analysis is complete, click Next.
6. Type the password of the Cluster Service Account and click Next.
7. Review the Proposed Cluster Configuration for accuracy. If the information is correct, click Next.
8. When the Add Nodes Wizard is complete, click Next and then click Finish.
9. Close the Cluster Administrator.

Installing MSDTC as a SQL Server Failover Clustering Prerequisite

As discussed earlier in the chapter, MSDTC is required as a prerequisite for installing SQL clustering. To create the MSDTC resource, follow these steps:

1. Click Start, All Programs, Administrative Tools, Cluster Administrator.
2. Right-click the cluster resource, choose New, and then click Resource.
3. In the New Resource dialog box, enter **`Microsoft Distributed Transaction Coordinator`** for the name, select a resource type of Distributed Transaction Coordinator, and assign it to the cluster group similar to what is shown in Figure 18.2. Click Next.
4. Verify all nodes are listed as potential owners and click Next.
5. Add the storage resource that will host the Distributed Transaction Coordinator and choose the options to add the storage resource as a dependency as well as the cluster network name. Then click Next.

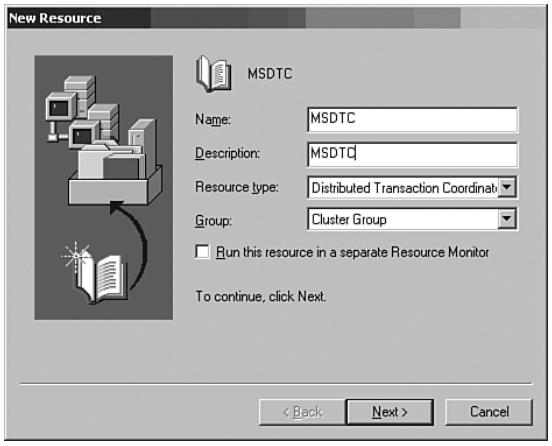

FIGURE 18.2
Creating a Distributed Transaction Coordinator.

Installing Failover Clustering for the Database Engine

Follow the next set of steps to install failover clustering for the first instance of SQL Server 2005. The components installed for this virtual instance are strictly for the Database Engine. These steps should be conducted on the first node of the Windows Cluster. Validate that the first node (TOR-CL01) is the owner for all cluster groups; then do the following:

1. Insert the SQL Server 2005 disk (Standard or Enterprise Edition).
2. Autorun should launch a splash screen with options for installing the prerequisites and application. (If autorun does not work, select Start, Run. Then type **`CDDrive:\servers\splash.hta`** and click OK.)
3. On the Start page, click Server Components, Tools, Books Online, and Samples to install SQL Server 2005, as illustrated in Figure 18.3.
4. On the license agreement page, click I Accept the Licensing Terms and Conditions and then click Next.

Note

Before you can install Microsoft SQL Server 2005, the Setup Installation Wizard verifies whether the necessary prerequisites such as Microsoft SQL Native Client and Microsoft SQL Server 2005 Setup Support Files have been installed. If the prerequisites have not been met, they are installed now.

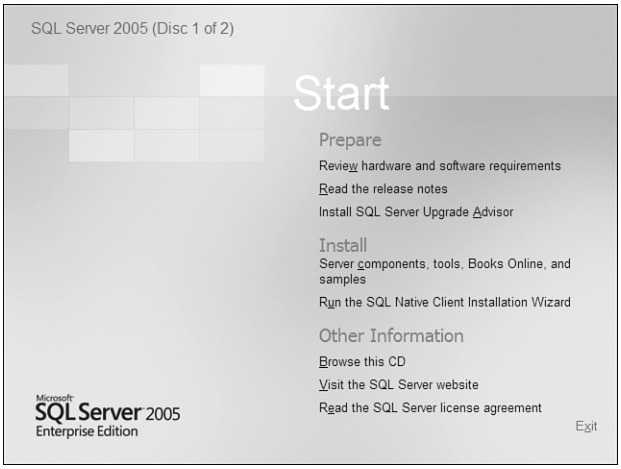

FIGURE 18.3
SQL Server 2005 installation splash screen.

5. On the SQL Server Component Update page, click Install to install the necessary prerequisites for SQL Server 2005. To continue after the required components are installed, click Next.
6. On the Welcome to the Microsoft SQL Server Installation Wizard page, click Next.
7. On the System Configuration Check (SCC) page, the wizard verifies that the system does not have any installation problems and provides success or failure status, including a message on each action being analyzed. To proceed with Setup after the SCC scan is complete, click Next, as illustrated in Figure 18.4.
8. On the Registration Information page, personalize the installation by entering name and company information; then click Next.
9. On the Components to Install page, select SQL Server Database Services; then select Create a SQL Server Failover Cluster. Optionally, select Workstation Components for managing the SQL Server cluster and then click Next.

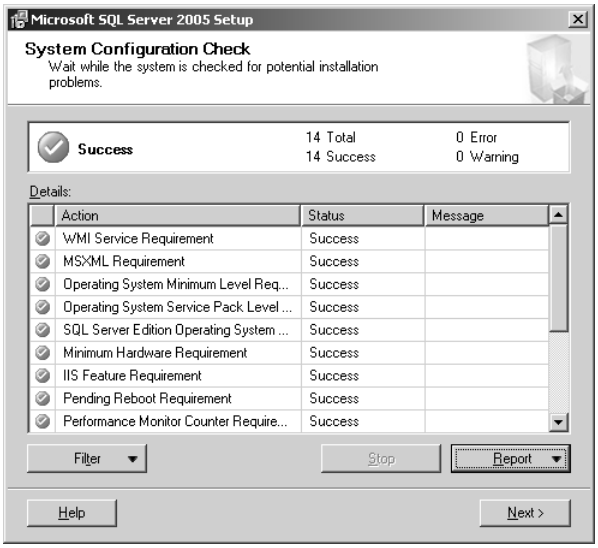

FIGURE 18.4
System Configuration Check screen.

Note

When installing failover clustering, you must use a valid local disk drive on all cluster nodes for the program location. This is typically the C: drive and not a shared cluster disk.

10. On the Instance Name page, select a default or named instance for the SQL Server cluster installation and then click Next. For this example, use Instance01 as the named instance.
11. Enter the name of the virtual server on the Virtual Server Name page, as shown in Figure 18.5, and click Next. For this example, use TOR-SQL01.

Note

Each virtual server name used for failover clustering must be unique on the network.

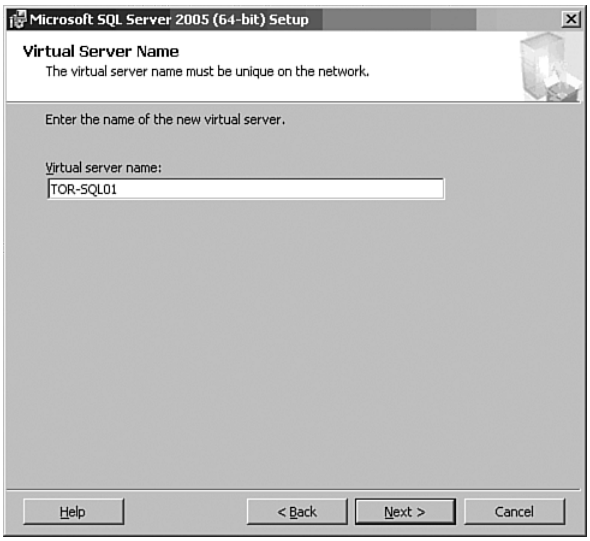

FIGURE 18.5
Enter the name of the virtual server.

12. On the next Virtual Server Configuration screen, enter the network that will be used and a dedicated static IP address. For this example, use the dedicated public network and IP address 192.168.115.206, as illustrated in Figure 18.6. After entering the virtual server information, click Add and then click Next to continue.
13. On the Cluster Group Selection page, select the cluster group where the virtual server resources will be installed. For this example, choose the Instance01 cluster group. In addition, after selecting the cluster group, verify or change the default data files location, as shown in Figure 18.7. Then click Next.

Note

When selecting a cluster group for the virtual instance installation, do *not* select the quorum drive. You should use a dedicated cluster group. In addition, each SQL Server virtual instance requires a dedicated cluster group with dedicated resources.

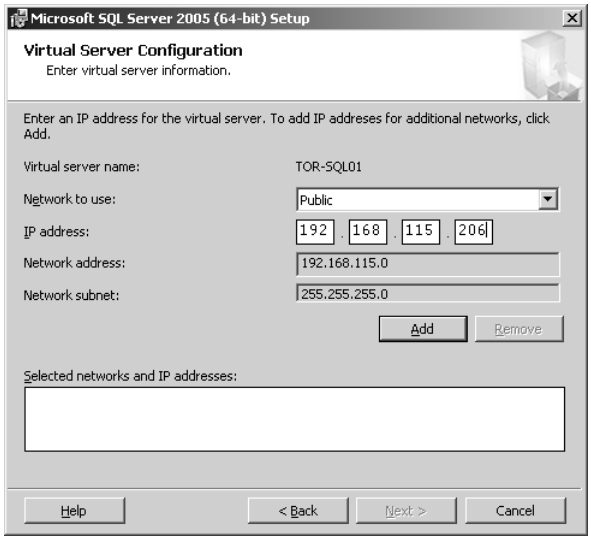

FIGURE 18.6
Virtual Server Configuration screen.

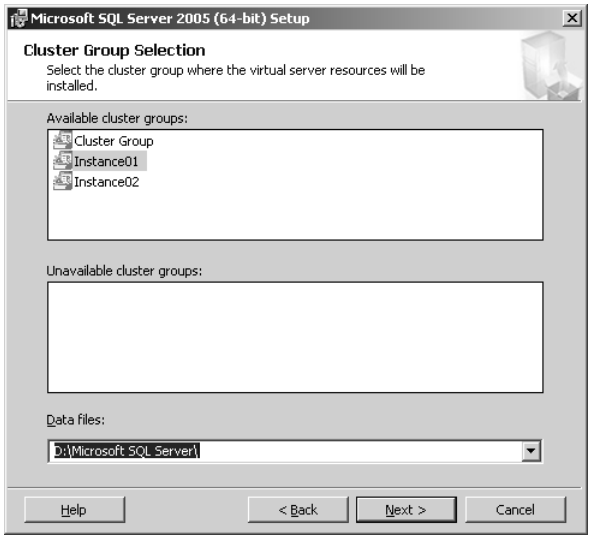

FIGURE 18.7
Selecting the virtual server cluster group.

14. On the Cluster Node Configuration page, select the nodes to be included in the SQL Server cluster configuration, as shown in Figure 18.8, and click Next.

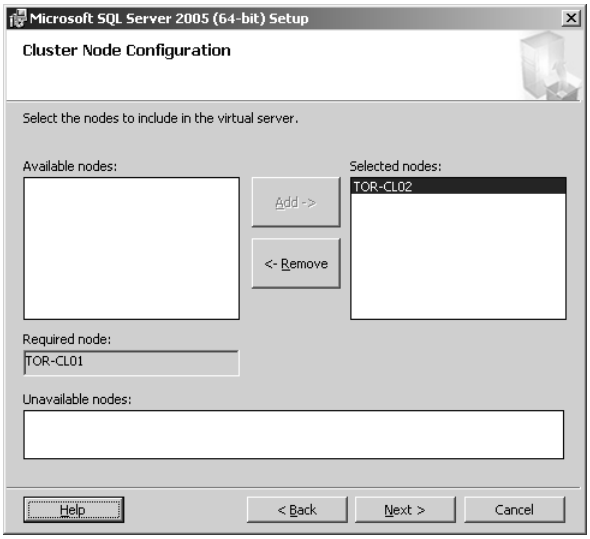

FIGURE 18.8
Selecting the nodes to be associated with the virtual server.

Note

When installing clusters, database administrators typically choose all the available nodes within the Windows cluster to be associated with the virtual instance of SQL Server. This choice means the SQL Server virtual instance can fail among each of the nodes selected. Sometimes, however, you want to select only a few nodes of the cluster to be associated with a SQL Server virtual instance. For example, if there is an eight-node cluster, nodes 1 to 4 can be dedicated for a specific instance, whereas nodes 5 to 8 can be dedicated to another instance.

15. On the Remote Account Information page, enter a username and password for a valid administrator account that has administrative privileges on all nodes of the cluster and click Next.

Note

Most likely, the Account Name box in the Remove Account Information page is already prepopulated by the user account that logged in to the server prior to the start of the SQL Server failover installation. If the account is not populated, enter a valid account and credentials.

16. On the Service Account page, enter a valid domain user account name, password, and domain name, as shown in Figure 18.9, and click Next.

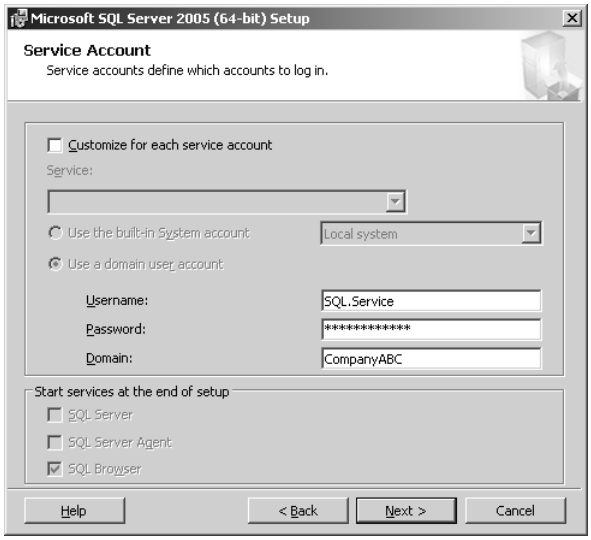

FIGURE 18.9
Inputting the SQL cluster service account.

17. Enter the appropriate existing domain group for each clustered service, as shown in Figure 18.10. Click Next to continue.

Note

SQL Server 2005 requires a domain group for each clustered server service installed. These groups must exist before the installation. Each service should use a separate, dedicated domain group. These domain groups should not be shared or used for any other purposes.

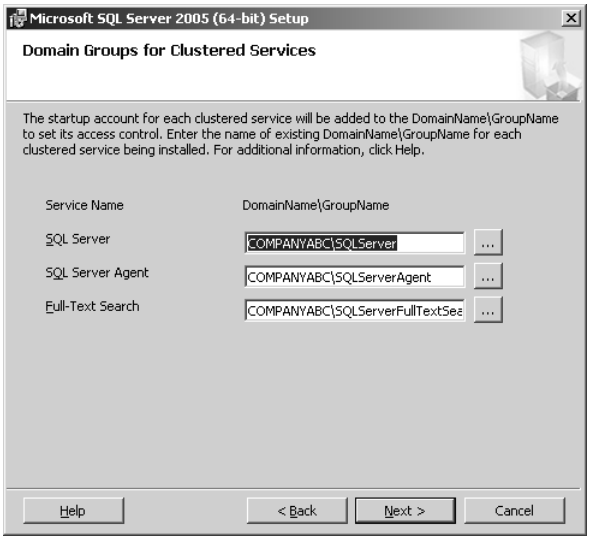

FIGURE 18.10
Inputting the startup account for each clustered service.

18. On the Service Account page, select either Use the Built-In System Account or Use a Domain User Account. In addition, place a check on the services that will start automatically. Alternatively, specify a dedicated account for each service. Click Next.
19. On the Authentication Mode page, select the authentication mode to use for the installation and then click Next.
20. On the Collation Settings page it is possible to change default collation settings used by the Database Engine and Analysis Services for language and sorting purposes. In addition, collation designator and sort order settings can be modified for each service account. Click Next to continue.
21. On the Error and Usage Report Settings page, choose whether to participate in the Customer Experience Improvement Program and then click Next. If you choose to participate, error reports and feature usage data for SQL Server 2005 will automatically be sent to Microsoft.
22. Before commencing the SQL Server 2005 installation, review the components that will be installed on the Ready to Install page and then click Next.

23. When the installation process starts, you can monitor its progress accordingly. New to SQL Server 2005 Setup is the ability to see the progress of each node during clustered installs. To view the installation status of another node, select it from the Node list.
24. On the Setup Summary page, click Next.
25. To exit the SQL Server Installation Wizard, click Finish on the Completing Microsoft SQL Server 2005 Setup page.

Note

In the Cluster Administrator, verify all the new SQL Server resources have come online after a successful installation. If the resources did not come online, reboot each node in sequence and check again.

Installing Failover Clustering for Analysis Services

Follow the steps in this section to install failover clustering for Analysis Services. Based on the online analytical processing (OLAP) cube storage model, this installation of Analysis Services requires a dedicated Database Engine to store the relational databases for reporting purposes. Therefore, this example includes the Analysis Services and Database Engine components as a separate installation. The second node in the cluster is the primary node for this installation.

Conduct the following steps on the second node of the Windows Cluster (TOR-CL02):

1. Using the Cluster Administrator, fail all the cluster groups to the second node by right-clicking each cluster group and selecting Move Group. For this example, the cluster group names are Cluster Group, Instance01, and Instance02.
2. Insert the SQL Server 2005 disk (Standard or Enterprise Edition) and repeat steps 2 through 8 in the preceding section "Installing Failover Clustering for the Database Engine."
3. On the Components to Install page, select Analysis Services and Database Engine and then select Create an Analysis Server Failover Cluster, as shown in Figure 18.11. Click Next to continue.
4. On the Instance Name page, select a named instance for the SQL Server cluster installation and then click Next. For this example, the named instance is Instance02.

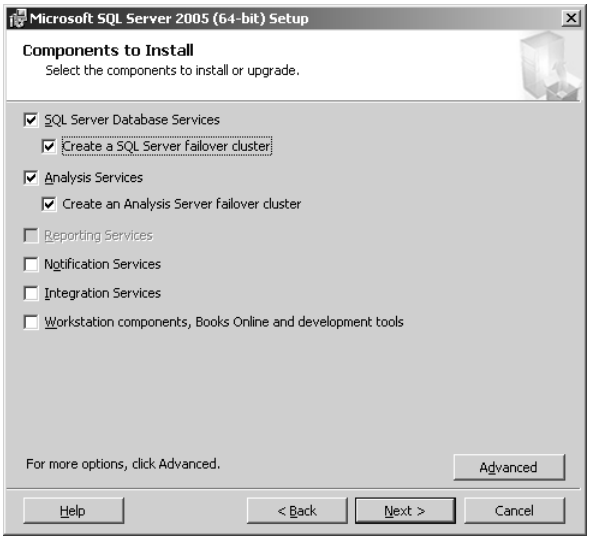

FIGURE 18.11
Selecting the Analysis Services cluster components.

5. Enter the name of the virtual server on the Virtual Server Name page and click Next. For this example, use TOR-SQL02.

> **Note**
> The virtual server name must be unique on the network.

6. On the next Virtual Server Configuration screen, enter the network that will be used and a dedicated static IP address. For this example, use the dedicated public network and IP address 192.168.115.207. After entering the virtual server information, click Add and then Next to continue.
7. On the Cluster Group Selection page, select the cluster group where the virtual server resources will be installed. For this example, choose the Instance02 cluster group. In addition, after selecting the cluster group, verify or change the default data files location, as shown in Figure 18.12. Then click Next.

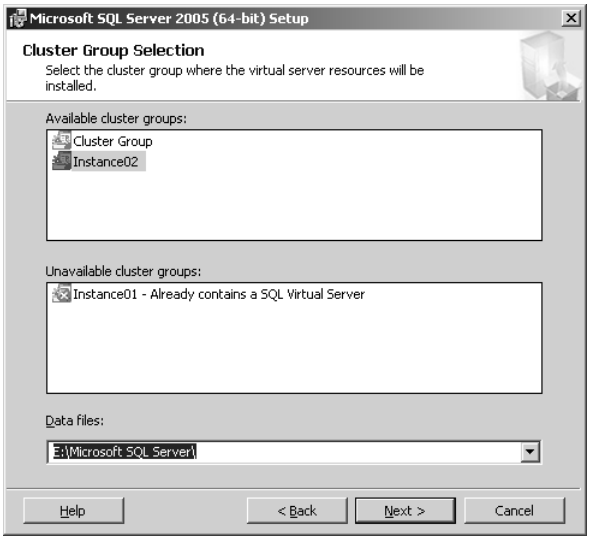

FIGURE 18.12
Selecting the desired clustered group for the Analysis Services installation.

Note

In the Cluster Group Selection page, shown in Figure 18.12, notice that the first cluster group, Instance01, is unavailable because it already contains a SQL virtual instance.

8. In the Cluster Node Configuration page, select the nodes to be included in the SQL Server cluster configuration and click Next.
9. Enter the appropriate existing domain group for each clustered service, and then click Next.
10. On the Service Account page, select either Use the Built-In System Account or Use a Domain User Account. In addition, check the services that should start automatically. Alternatively, specify a dedicated account for each service. Then click Next.
11. On the Authentication Mode page, select the authentication mode to use for the installation and then click Next.
12. On the Collation Settings page you can change default collation settings used by the Database Engine and Analysis Services for language and sorting purposes. In addition, collation designator and

sort order settings can be modified for each service account. Click Next to continue.

13. On the Error and Usage Report Settings page, select whether to participate in the Customer Experience Improvement Program and then click Next. If you choose to participate, error reports and feature usage data for SQL Server 2005 will automatically be sent to Microsoft.
14. Before commencing the SQL Server 2005 installation, review the components that will be installed on the Ready to Install page and then click Next.
15. When the installation process starts, monitor its progress. New to SQL Server 2005 Setup is the capability to see the progress of each node during clustered installs. To view the installation status of another node, select it from the Node list.
16. On the Setup Summary page, click Next.
17. To exit the SQL Server Installation Wizard, click Finish on the Completing Microsoft SQL Server 2005 Setup page.

Administering Patch Management on a SQL Server 2005 Failover Cluster

Similar to a traditional nonclustered SQL Server, the operating system and SQL Server application require ongoing patch management to keep the servers up to date. Patch management includes installing service packs and critical hotfixes for both the operating system and SQL Server. When you're working in a clustered environment, each node within the cluster should have the exact same service pack and hotfixes to ensure consistency.

One of the main benefits of using Windows clusters is your ability to install software, service packs, and critical updates on a node without interrupting service of the cluster. This process is known as a *rolling upgrade*. For example, when you install Windows Server 2003 Service Pack 2, all the cluster groups can be failed over to the second node, and the installation can then be conducted on the first node without affecting client operations. The node can be rolled back to node 1, and Windows Server 2003 Service Pack 2 can be applied to the second node.

Installing Windows Service Packs on a Failover Cluster

Ongoing Windows patch management is a reoccurring monthly affair that plagues organizations. Fortunately, when you're working with clusters, it is

possible to apply service packs and hotfixes in a rolling fashion without causing server downtime.

Follow these high-level steps when applying patches to a cluster:

1. From the primary node, use the Cluster Administrator and pause the primary node.
2. Move all the cluster groups from the primary node to a failover node.
3. Install the service pack or hotfix on the primary node and then restart the node.
4. When the primary node restarts, resume cluster operations by selecting Resume on the primary node while using the Cluster Administrator.
5. Repeat this process for all additional nodes within the cluster.

The goal of the rolling upgrade process is to pause the primary node on which the installation will occur, move all cluster groups off the primary node, and then install the service pack or hotfix. This process prevents service interruptions because the application is always available.

Installing SQL Server Service Pack 2 on a Failover Cluster

Unlike with an operating system service pack such as Windows Server 2003 Service Pack 1, it is not possible to conduct a rolling upgrade of SQL Server 2005 Service Pack 2 when upgrading a failover cluster instance. Therefore, planned downtime is required and should be scheduled when applying Service Pack 2 because SQL Server will be unavailable during the installation. In addition, a server outage is inevitable because each node needs rebooting after the installation is complete.

> **Note**
>
> If SQL Server management tools were installed manually on nodes other than the primary node where the original installation was started, these components need to be patched manually because they will not be affected by the service pack install.

The Service Pack 2 installation is quite intuitive—so much so that it recognizes all components installed and whether more than one instance of SQL Server is installed within the failover cluster. If multiple instances of SQL Server are running, it is possible to streamline the installation and reduce downtime because a single SQL Server service pack installation process can

upgrade each of the virtual servers found within the cluster at once. If the instance is owned by another node, the installer may not recognize the SQL Server virtual instance, and the service pack may need to be applied separately for each instance.

Note

If you want to minimize downtime and duplication of service pack installations, it is a best practice to move the cluster groups of all virtual servers to the primary node before running the installation. If you don't do this, you have to install SQL Server Service Pack 2 separately on each primary node because the installer needs exclusive access to all shared resources.

The following example illustrates an installation of SQL Server 2005 Service Pack 2 on a failover cluster that is running two instances of the Database Engine and one Analysis Services instance.

Note

It is a best practice to back up the full SQL Server installation before conducting a service pack upgrade. This includes backing up the user and system databases. For more information on backups, see Chapter 17, "Backing Up and Restoring the SQL Server 2005 Environment" (online).

To update SQL Server 2005 with Service Pack 2, download the appropriate service pack binaries from the Microsoft website (www.microsoft.com/download) and follow these steps:

1. Conduct a full SQL Server backup for every instance and component installed within the failover cluster.
2. Using the Cluster Administrator, fail each cluster group that is owned by the second node over to the first node by right-clicking each cluster group and selecting Move Group.
3. Before starting the Service Pack 2 installation, verify all cluster resources are online and owned by the first node in the Cluster Administrator, as shown in Figure 18.13.
4. Start the SQL Server Service Pack 2 installation by double-clicking on the downloaded file.
5. On the Welcome screen, as shown in Figure 18.14, click Next to continue.

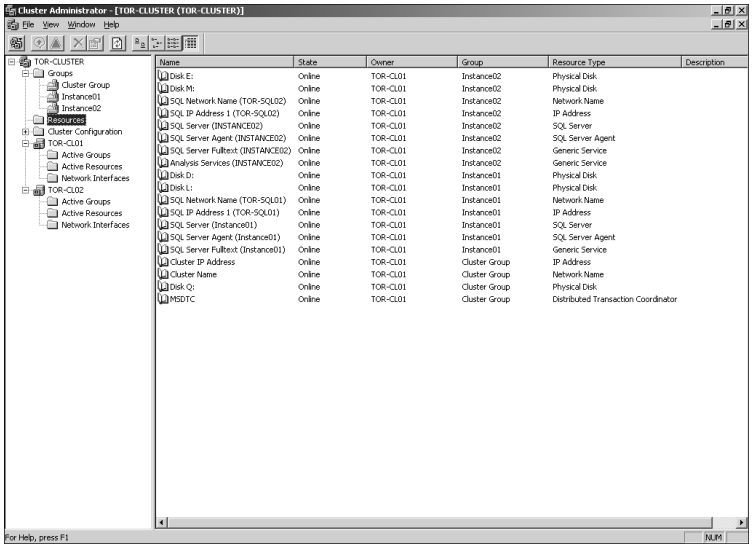

FIGURE 18.13
Verifying the owner of each cluster resource.

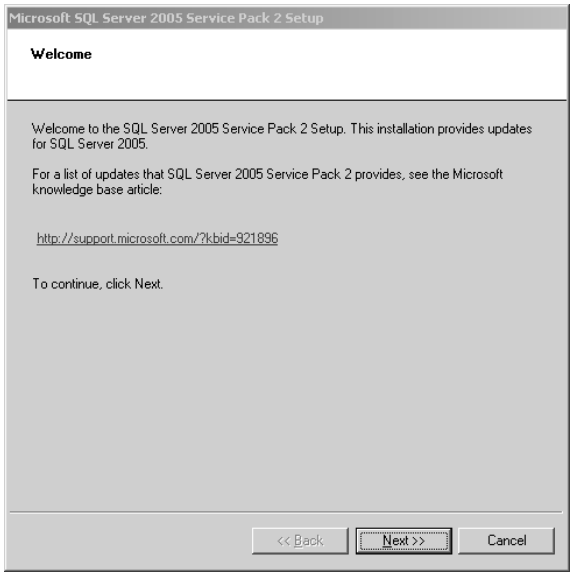

FIGURE 18.14
Microsoft SQL Server 2005 Service Pack 2 Setup Welcome screen.

6. Read the licensing agreement. Select I Agree if you agree with the terms. Click Next to continue.
7. On the Feature Selection page, select the program features that should be upgraded to Service Pack 2, as shown in Figure 18.15, and click Next to continue.

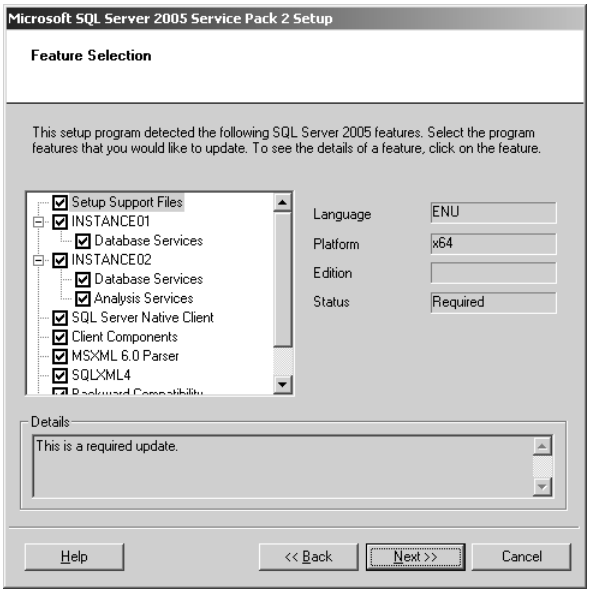

FIGURE 18.15
Microsoft SQL Server 2005 Service Pack 2 Setup Feature Selection screen.

8. For this example, notice how both Virtual Server instances (Instance01 and Instance02) are detected and available for the upgrade. The reason is that the primary node owns all the cluster resources. Instances that are not owned by the primary node would not be available and would have to be upgraded at a separate time.
9. On the Authentication page, as shown in Figure 18.16, select the Authentication mode, either Windows Authentication or SQL Authentication, that the setup program will use to install SQL Server Service Pack 2. Click Test to verify the installation account credentials for each instance being upgraded and click Next to continue.

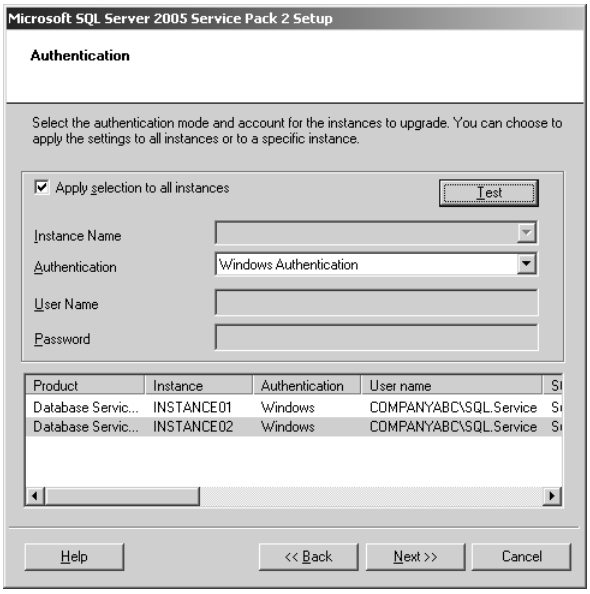

FIGURE 18.16
Microsoft SQL Server 2005 Service Pack 2 Setup Authentication screen.

Note

If the authentication mode, username, and password are the same for all components or instances being updated, select the Apply Selection to All Instances check box. Otherwise, provide the instance name, authentication type, username, and password information for each component or instance being updated.

10. If the account is not prepopulated on the Remote User Account page, as shown in Figure 18.17, enter the remote user account name and password. This account requires administrator privileges for all nodes of the failover cluster. Therefore, the account should be the Cluster Administrator account and not the SQL Server Administrator account. Click Next to continue.
11. On the Error and Usage Report Settings page, select whether to participate in the Customer Experience Improvement Program and then click Next. If you choose to participate, error reports and feature usage data for SQL Server 2005 will automatically be sent to Microsoft.

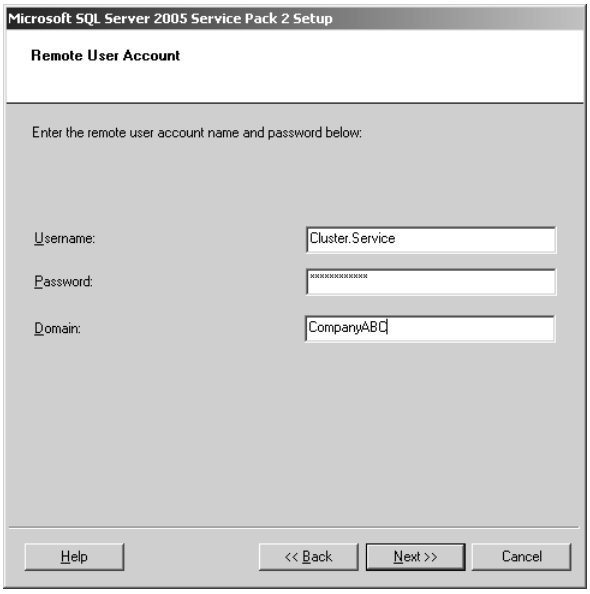

FIGURE 18.17
Entering remote user account credentials.

Note

Participating in the Customer Experience Improvement Program is generally a good idea because the information sent to Microsoft assists the company with building a stronger product and addressing customer errors. Personal data is not typically sent to Microsoft; however, you should read the warnings thoroughly because error reports may unintentionally include personal information from time to time. Microsoft states it will not use the personal information captured.

12. On the Running Processes page, as shown in Figure 18.18, review the locked files identified by the wizard and either end the processes to avoid a computer restart or click Next to continue.

    Unlike with a standalone SQL Server instance, rebooting each node when upgrading SQL Virtual Service is a requirement. Therefore, it is senseless to end each locked process because the reboot is inevitable. As noted before, don't forget to schedule downtime with the end user community.

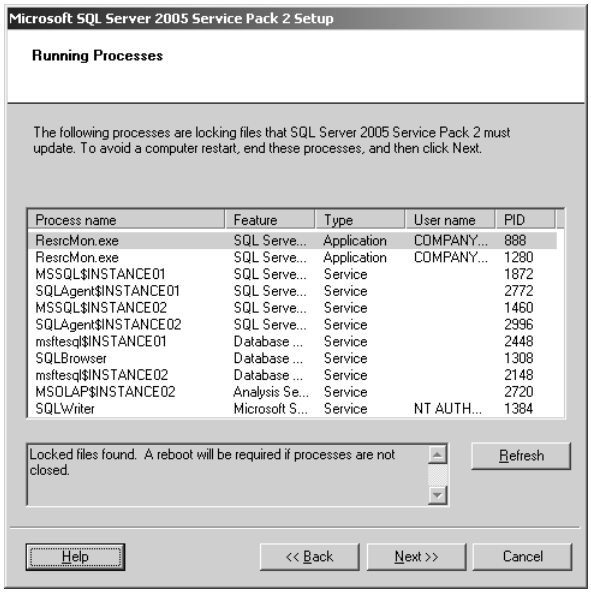

FIGURE 18.18
Microsoft SQL Server Service Pack 2 Setup Running Processes screen.

Tip

In production, it is not always possible or realistic to stop the processes responsible for locking SQL Server files because this undertaking results in a service outage. As a result, it is best practice to schedule planned downtime when production SQL servers will be updated with the latest service pack.

13. On the Ready to Install page, click Install to initiate the installation of SQL Server Service Pack 2.
14. On the Installation Progress page, verify the results of the installation and click Next to continue. Because this is a SQL cluster being upgraded, a reboot warning message appears.

    The Installation Progress page monitors the installation of Service Pack 2 and provides a success or failure message for each component being upgraded. The wizard also indicates whether a reboot is required. A reboot is required if files have operations pending.

15. For a summary of the installation, click View Summary on the Installation Complete page and then click Next.

    The summary results are displayed in Notepad and can be saved to a text file for future analysis. The Details section of this page also displays the location of each summary log file.

16. The final screen includes additional information pertaining to security and Vista (see Figure 18.19). Review the information on the Additional Information page and click Finish to complete the SQL Server 2005 Service Pack 2 installation.

Note

The final screen includes information pertaining to security and Vista, Microsoft's latest operating system. By default, Vista users who are also part of the Windows Administrators group are not automatically granted permission to connect to SQL Server. If these users need access to SQL Server, they need to be provisioned. If you're using Vista, follow the steps in the wizard to provision new Vista users by enabling the option Launch the SQL Server 2005 Provisioning Tool for Vista to grant administrative rights to the appropriate Windows Vista users.

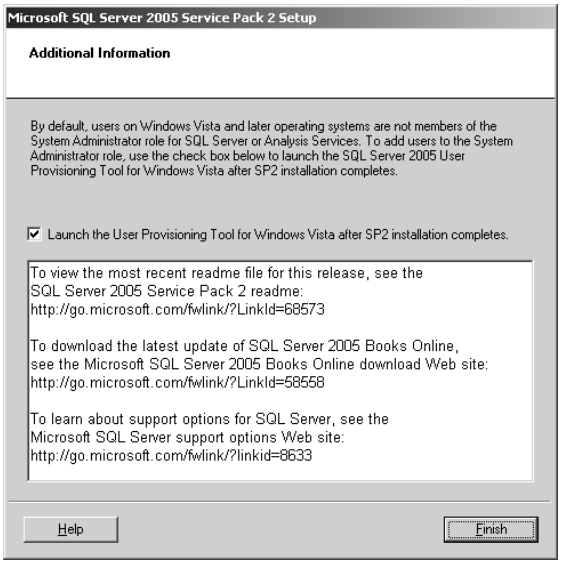

FIGURE 18.19
Microsoft SQL Server Service Pack 2 Setup Vista information screen.

17. Reboot the first node and then the second node.
18. Verify the SQL Server and Windows logs for any errors.

Managing a SQL Server 2005 Failover Cluster

The following sections focus on management tasks after a SQL Server 2005 failover cluster has been implemented. These tasks are in no particular order.

Adding Additional Nodes to an Existing SQL Server Failover Cluster

Because the SQL Server 2005 Enterprise Edition supports only up to eight nodes within a cluster, sometimes you may need to add an additional node to an existing SQL Server 2005 cluster instance. Unfortunately, to achieve this goal, you must rerun Setup and add the new node to the failover cluster instance because the SQL Server installation needs to place specific files on all the nodes within the cluster. You can change Setup by using Add or Remove Programs in the Control Panel, selecting the SQL Server component, and then selecting Change.

Managing Additional Cluster Drives

A SQL Server virtual instance uses only one shared hard drive during the installation of SQL Server failover clustering. Therefore, even though additional share drives are available in the cluster, such as the log and backup drives, a SQL Server instance cannot leverage those shared drives unless each additional shared drive is added as a resource dependency for the SQL Server name clustered resource in the Cluster Administrator. Unfortunately, you must take the cluster group offline when adding additional shared disk resources and do the following:

1. Take the cluster group offline.
2. In the Cluster Administrator, expand the Groups folder and select Instance01.
3. In the right pane, right-click the SQL Server Name resource, such as SQL Server Instance01, and select Properties.
4. On the Dependencies tab, select Modify. Then in the Modify Dependencies window, add the desired drives from the Available Resources list, as shown in Figure 18.20.
5. Click Apply and then OK and bring the cluster group back online.

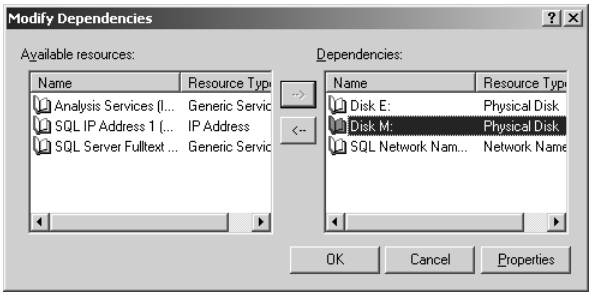

FIGURE 18.20
Managing cluster resource disk dependencies.

Run the following Transact-SQL syntax to verify that the newly added resource dependency drives are available to SQL Server:

```
SELECT * FROM ::FN_SERVERSHAREDDRIVES()
```

Removing SQL Server Failover Clustering

Similar to adding a new node to an existing cluster, you also need to launch the SQL Server 2005 installation to remove a single instance or all instances within the cluster. The steps to remove an instance within the cluster consist of using Add or Remove Programs in the Control Panel, selecting the SQL Server 2005 component to uninstall such as Microsoft SQL Server 2005, and then selecting Remove.

Managing SQL Server Failover Service Accounts

Sometimes a service account or a service account password needs to be changed for the failover cluster instance. Similar to a standalone SQL Server installation, all SQL Service account changes should be conducted with the SQL Server 2005 Configuration Manager tool. In addition, when you're working with clusters, all nodes must be online when a service account changes. As a reminder, when allocating service accounts, follow the practice of least privilege and isolation; as a result, the SQL Cluster Service Account should be a different account from the service account running the Windows cluster.

Managing Cluster Group Failovers

To manage cluster group failovers, follow these steps:

1. From the first node (TOR-CL01), choose Start, All Programs, Administrator Tools and Cluster Administrator.
2. In the Open Connection to Cluster screen, either type in the name of the cluster, or click Browse to locate the server and then click OK. For this example, use the TOR-CLUSTER.
3. Expand Groups, right-click the default cluster group, and click Move Group. The cluster group is moved from node 1 (TOR-CL01) to node 2 (TOR-CL02).
4. In the right pane, verify the owner of the cluster group is node 2 (TOR-CL02).
5. Repeat the process to return the cluster group to node 1 or use the same steps to move any other desired groups such as Instance01 or Instance02.

Managing the Quorum Resource Log Size

The Quorum resource maintains a log file. By default, the quorum log is stored on the Quorum shared disk, and the log file is reset at 4096KB. For SQL Server clusters, it is a best practice to increase the default quorum log reset interval to a larger size due to the number of file shares and transactions. If the log size is not large enough, it is common for resources to be negatively affected when coming online. A typical rule to follow when sizing the quorum log file is to double the size of the Clusdb file. You can locate the Clusdb file on each node in the %SystemRoot%\Cluster folder. Follow these steps to resize the quorum log:

1. In the Cluster Administrator, right-click the cluster name (TOR-CLUSTER).
2. Select the Quorum tab.
3. Enter the desired reset value for the quorum log such as 8192KB, as shown in Figure 18.21, and then click OK.

Verifying the Status of Clustered Groups and Resources

As database administrator, you frequently need to know which node is the owner of all the cluster groups and resources. The resource folder in the Cluster Administrator displays all resources, states, owners, groups, and resource types within the cluster, regardless of which cluster group the resource resides in.

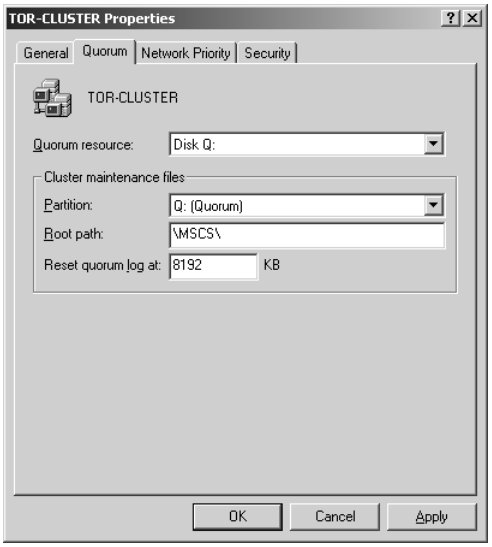

FIGURE 18.21
Managing the quorum root path and log size.

To verify the resource status of a cluster, follow these steps:

1. In the Cluster Administrator, click on the Resource folder.
2. View the right pane, which depicts the status, owner, group, and resource type for each resource, as shown in Figure 18.22.

Managing Preferred Owners of a Cluster Node and Group

For each cluster group, you can assign a preferred owner. In the example, there are two instances of SQL Server installed within the cluster. From a performance perspective, it is possible to configure node 1 to be the preferred owner of TOR-SQL/Instance01 and node 2 to be the preferred owner of TOR-SQL02/Instance02. Therefore, when the servers are brought online, the preferred owners maintain service operations of the desired cluster group and SQL Server virtual instance. It is worth mentioning that preferred owners are necessary when you are running more than two nodes within a cluster. The preferred owners list dictates the failover behavior to the next available node based on the nodes in the list. For example, in an eight-node cluster, you may have dedicated passive standbys by having node 1 first failover to node 3, then node 4, and then node 5.

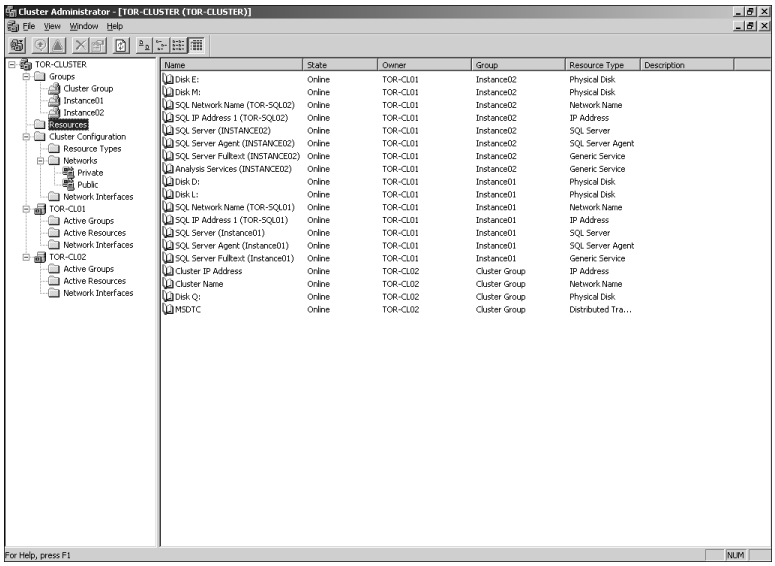

FIGURE 18.22
Verifying cluster resource ownership, state, and group membership.

Tip

When the cluster nodes are turned on, the Cluster Administrator tries to assign the SQL Server Virtual Server to the first node in the preferred owners list. If this node is not available, the server is assigned to the next server name in the list. This behavior is similar to the failover process. If a failover occurs, the SQL Server cluster fails over to the available passive nodes based on the preferred owners list.

Follow the next sets of steps to configure preferred owners for the multiple instances installed in the SQL Server cluster example. Instance01 is homed in on node 1 (TOR-CL01), and Instance02 is homed in on node 2 (TOR-CL02). Follow these steps for cluster group Instance01:

1. In the Cluster Administrator, expand the Groups folder and select Instance01.
2. Right-click Instance01 and select Properties.
3. On the General tab, verify TOR-CL01 is the only server listed in the Preferred Owners tab. If not, click the Modify tab and make the appropriate changes. Click Apply and then click OK.

Follow these steps for cluster group Instance02:

1. In the Cluster Administrator, expand the Groups folder and select Instance02.
2. Right-click Instance02 and select Properties.
3. On the General tab, verify TOR-CL02 is the only server listed in the Preferred Owners tab. If not, click the Modify tab and make the appropriate changes. Click Apply and then click OK.

Managing Failover Clustering Failover Policies

There are a few ways to control what will trigger a failover or how a failover will be triggered. First, a global setting controls the threshold and period of a failover. This setting affects the whole cluster group and dictates how many times a failover can occur during a period of time. Second, advanced parameters can be configured on each individual resource. Advanced parameters dictate whether the individual resource failure will affect the whole cluster group and intervals settings such as Looks Alive and Is Alive.

Follow these steps to configure failover settings for a cluster group such as Instance01. This example uses three failovers within a three-hour period:

1. In the Cluster Administrator, expand the Groups folder and select Instance01.
2. Right-click Instance01 and select Properties.
3. On the Failover tab, set the Threshold setting to 3 and the Period in hours to 3, as illustrated in Figure 18.23.
4. Click Apply and then click OK.

Managing Failover Clustering Failback Policies

When a primary node fails, you can control the behavior of the failback after the primary node becomes active again. The failback can be set to immediately, between a specific time of the day, or can be prevented. For mission-critical production clusters, it is a best practice to either prevent automatic failback or set the failback during hours of nonpeak usage. By using these settings, you can fail back the node manually or during nonpeak usage. As a result, the end user community and application are not affected, resulting in downtime when the node fails back.

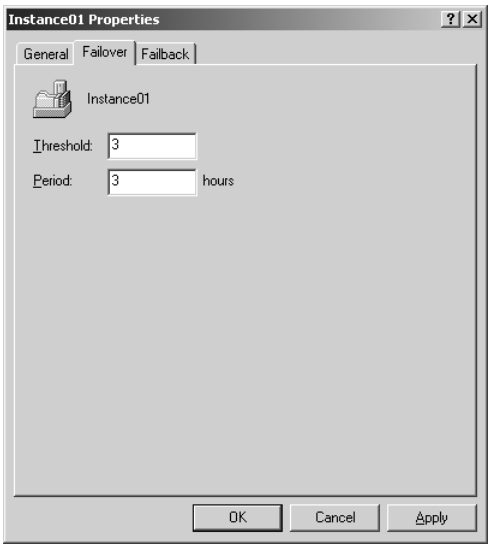

FIGURE 18.23
Managing cluster resource failover settings.

Follow these steps to configure failback settings for a cluster group such as Instance01:

1. In the Cluster Administrator, expand the Groups folder and select Instance01.
2. Right-click Instance01 and select Properties.
3. On the Failback tab, select the option to Failback Between 1 and 2 in hours, as shown in Figure 18.24.
4. Click Apply and then click OK.

Managing Kerberos Authentication and DNS Registration for Network Names

Each network name for the cluster, such as the Windows cluster name or a SQL Server instance name, shows up in Active Directory Users and Computers as a computer. These computers can be placed into Active Directory organization units for management, administration, and application of Active Directory Group Policies.

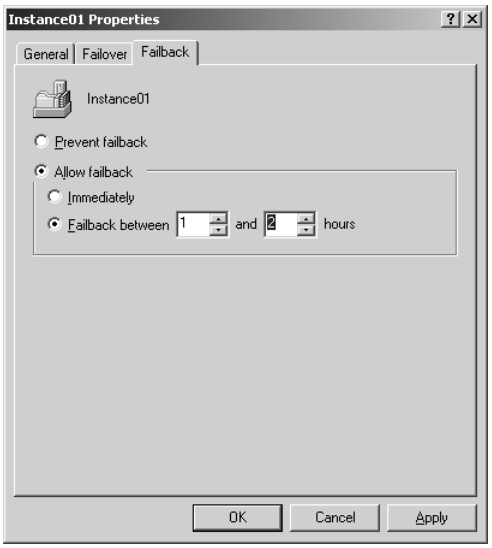

FIGURE 18.24
Managing cluster resource failback settings.

The network name also has an advanced parameter setting that ensures that, before the cluster resource comes online, the A record for the virtual server has been updated in the domain name service (DNS). If the virtual server does not have an appropriate A record in DNS, the resource fails.

Another advanced setting involves enabling Kerberos authentication for a virtual server. Kerberos authentication is much more secure than the traditional NTLM authentication. Authentication of the virtual server is required because it acts like a computer integrating with an Active Directory domain.

Follow these steps to enable the DNS Registration Must Succeed and Enable Kerberos Authentication options on a cluster virtual name:

1. In the Cluster Administrator, expand the desired cluster group folder and select the desired network name resource.
2. Right-click the network name resource and select Properties.
3. On the Parameters tab, select both DNS Registration Must Succeed and Enable Kerberos Authentication, as illustrated in Figure 18.25.
4. Click Apply and then click OK.
5. Repeat these steps for all network name resources within the cluster.

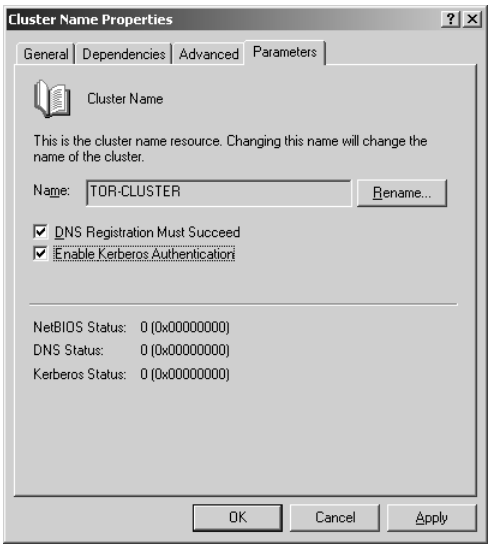

FIGURE 18.25
Managing advanced cluster name properties.

Pausing and Resuming a SQL Server Cluster Node

When you're conducting maintenance or rolling upgrades, it is common to pause a node. When a node is paused, the existing cluster groups and resources stay online; however, additional groups and resources cannot be brought online. Follow these steps to pause and resume a SQL Server cluster node:

1. In the Cluster Administrator, click on the desired node to pause such as TOR-CL01.
2. Right-click the node and select Pause.
3. Verify there is an exclamation mark next to the node, indicating it has been paused.
4. Repeat the preceding steps to resume the cluster, but in step 2 select Resume instead of Pause.
5. Verify that the exclamation mark is no longer present, indicating the node has been resumed.

Adding New Windows Nodes to a Cluster

To add new nodes to a cluster, perform these steps:

1. Open the Cluster Administrator.
2. Right-click the cluster name, choose New, and then click Node.
3. When the Add Nodes Wizard launches, click Next.
4. Enter the name of the system that will be added to the cluster, click Add, and then click Next.
5. The Analyzing Configuration screen launches. When its analysis is completed and successful, click Next.
6. When the node has been added, click Finish.

Renaming Clustered Resources

To rename a resource in a cluster, perform these steps:

1. Open the Cluster Administrator.
2. Expand the cluster name.
3. Click Resources in the left pane.
4. Right-click a resource in the right pane and choose Rename.
5. Type a new name for the resource and press Enter.

Renaming Clustered Groups

To rename a group in a cluster, perform these steps:

1. Open the Cluster Administrator.
2. Expand the cluster name.
3. Expand the Groups container.
4. Right-click a group in the left pane and select Rename.
5. Type a new name for the group and press Enter.

Summary

Failover clustering is a great high-availability alternative for maintaining maximum uptime for mission-critical databases and the whole SQL Server instance. Failover is seamless and transparent to end users and clients. The correct clustering topology must be selected based on Service Level Agreements, availability requirements, and budget.

Best Practices

- Before installing SQL Server failover clustering, understand the prerequisites and verify that the clustering hardware is supported and certified by both the hardware vendor and Microsoft.
- Use identical hardware for all nodes in the cluster. This includes processor, memory, and firmware.
- Configure Microsoft Clustering Services prior to SQL Server 2005 Clustering.
- Ensure disk drive letters are identical on all nodes within the cluster.
- Avoid having the quorum resource and other cluster resources from sharing the same disk.
- Disable NETBIOS on the private/heartbeat network adapters.
- Disable write-back caching on host controllers.
- Do *not* configure dynamic disks because clustering supports only basic disk configurations.
- Determine whether a single-instance or multiple-instance configuration will be implemented. Plan the disk layout accordingly while taking future growth into account.
- Identify which SQL Server components will be installed.
- Do *not* use the same service account for Windows clustering and SQL Server clustering.
- Configure dependencies for shared disks so that they can be recognized and leveraged by SQL Server 2005.
- Before using multiple instances, understand the impact of multiple-instance configurations and performance degradation on the surviving node if a failover occurs.
- Change the service accounts only via SQL Server Configuration Manager.
- Do not configure MSDTC resources within the same cluster group as the SQL Server virtual instances.
- Use Add and Remove programs to modify a SQL Server failover cluster installation.
- Ensure that each virtual server name is unique on the Network/Active Directory domain.