Microsoft® SQL Server 2005

Failover Clustering for Microsoft SQL Server 2005 and SQL Server 2005 Analysis Services

SQL Server Technical Article

Writers: Allan Hirt (Avanade, Inc.) and Dave Whitney (Microsoft)

Project Editors: Louis Berner (Microsoft), Tresy Kilbourne (Microsoft)

Applies To: SQL Server 2005

Version 1.0

Summary: This white paper is intended mainly for a technical audience and not technical decision makers. It complements the existing documentation around planning, implementing, and administering of a failover cluster that can be found in Microsoft® SQL Server™ 2005 Books Online. To ease the upgrade process for existing users of failover clustering, this white paper will also point out differences in the failover clustering implementation of SQL Server 2005 compared to that of SQL Server 2000.

Copyright

Table of Contents

Introduction

When implementing Microsoft SQL Server, there are four basic tenets: availability, performance, manageability, and security. All four are linked, and one can affect the other. SQL Server has included availability features for quite some time, and one of those is the ability to create a *failover cluster*. Microsoft has made a serious commitment to database availability with SQL Server 2005, emphasized by the introduction of the SQL Server Always On Technologies program. Failover clustering is one of these technologies. Failover clustering is a mature availability option that has been a part of SQL Server since version 6.5; it was greatly enhanced with SQL Server 2000, and SQL Server 2005 continues that tradition.

This white paper complements the existing documentation about planning, implementing, and administering a failover cluster in SQL Server 2005 Books Online and provides many links to existing content. This white paper also points out differences between the failover clustering implementation of SQL Server 2005 and that of SQL Server 2000.

This paper is intended for a technical audience.

Enhancements in SQL Server 2005 Failover Clustering

The implementation of failover clustering in SQL Server 2005 includes many new features and enhancements:

- SQL Server 2005 Analysis Services can now be clustered.

- Both SQL Server 2000 and SQL Server 2005 can be installed in failover clustering configurations side-by-side on the same Microsoft Windows server cluster.

- SQL Server 2005 Standard Edition now supports failover clustering.

- Depending on your available resources (such as disk) and edition of SQL Server 2005, up to 25 clustered instances of SQL Server 2005 can be deployed in a cluster.

- Disk mount points are fully supported with failover clustering in SQL Server 2005.

- Windows on Windows (WOW) mode of 64-bit Windows Server® 2003 (x64) is supported for deploying 32-bit failover clusters.

- E-mail support in a clustered configuration is now possible via the new Database Mail feature of SQL Server 2005, which replaces the older SQL Mail.

- The SQL Server connectivity layer now depends on the new SQL Native Client, and not upon a specific version of Microsoft Data Access Components (MDAC).

- The SQL Server cluster resource DLL runs in its own resource monitor instead of sharing a default resource monitor for all instances.

- The installation process for a failover cluster can now be carried out unattended and fully scripted.

- Microsoft Virtual Server is supported for use with failover clustering.

- New dynamic management views (DMVs) are available for failover clustering.

- A failover clustering instance can be renamed without having to uninstall and reinstall SQL Server.

- Each installation of SQL Server 2005 in a clustered configuration now has its own full-text resource.
- The Service Master Key and Database Master Keys are distributed automatically across nodes.
- Enhanced debugging abilities have been built in for Microsoft Customer Service and Support.

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.

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.

Client Connections and Failover Clustering

End users and applications access a failover cluster with the SQL Server Network Name or IP address of the SQL Server failover cluster instance. The cluster name, the cluster IP address, and even the individual node names are not used by the connections. It does not matter to the client or application which node owns the resources because connecting to the instance appears as a normal instance of SQL Server.

During failover, 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. In any event, whatever the user was working on when the server went offline will likely not be completed/committed.

For more information, see Knowledge Base article 273673: <u>Virtual Server Client Connections Must be Controlled by Clients</u>.

Planning for Installation

Before running Setup to install SQL Server 2005, there are a few things that must be thought about, planned, and configured.

SQL Server and Windows Versions Supported

SQL Server 2005 can be installed only on Windows 2000 Server with Service Pack 4, or on an appropriate Windows Server 2003 with Service Pack 1 (or later) applied.

Important: The initial version (RTM) of Windows Server 2003 is not supported for SQL Server 2005, so you must plan on upgrading to Windows Server 2003 Service Pack 1 or later before attempting to install SQL Server 2005.

SQL Server failover clustering requires a Windows edition that supports clustering. These editions are Windows Server 2003 Enterprise Edition or Datacenter Edition, and Windows 2000 Advanced Server or Datacenter.

SQL Server 2005 failover clustering is only available in Enterprise Edition, Standard Edition, and Developer Edition including the evaluation editions of SQL Server 2005. Support for failover clustering in Standard Edition is new in SQL Server 2005, and only two-node failover cluster instances are allowed using that version of SQL Server, even though the cluster itself might have more nodes. Developer Edition cannot be deployed in production due to licensing restrictions, but has the same functionality as Enterprise Edition and can be deployed in development and test environments.

An excellent resource for all supported features and limits for SQL Server 2005 is [Features Supported by the Editions of SQL Server 2005](#).

Side-by-Side Installation with SQL Server 2000 Failover Clusters

You can install SQL Server 2005 failover clustering on an existing Windows failover cluster that has deployed one or more clustered instances of SQL Server 2000 if the cluster itself meets the minimum supportability requirements for SQL Server 2005 and you have enough capacity on your hardware to run more instances of SQL Server. The planning is akin to the preparation that you did when placing multiple SQL Server 2000 instances on the cluster – you must have at least one free dedicated and named disk available per instance, as well as a reserved and dedicated IP address and unique name for the instance in the domain. In a SQL Server 2005 side-by-side installation, SQL Server 2000 tools continue to manage SQL Server 2000, but all connectivity now is replaced by SQL Server Native Client. One benefit to a side-by-side install is that it creates a fallback plan in the event that something happens during the migration to SQL Server 2005.

Because there can only be one default instance on a stand-alone server or in a cluster, if one SQL Server instance is already configured as a default instance, whether local to a node or as a failover cluster instance, all instances of SQL Server 2005 must be deployed as named instances.

After installing SQL Server 2005, SQL Server 2000 Service Manager displays the node name instead of the virtual SQL Server 2005 server name. It does so because Service Manager was not coded to handle SQL Server 2005, and SQL Server 2005 does not include Service Manager. See Figure 1 for an example of SQL Server Service Manager and how a SQL Server 2005 clustered instance can be displayed. SQL Server Service

Manager has been replaced with SQL Server Configuration Manager in SQL Server 2005, and using it is described in the section "Starting, Stopping, and Pausing Services," later in this paper.
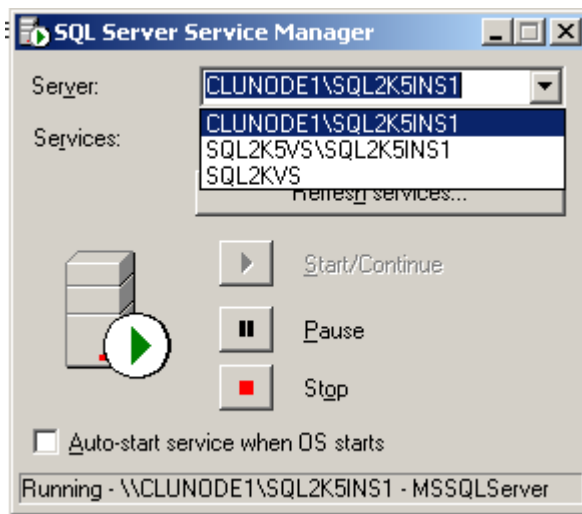
Figure 1. SQL Server 2000 Service Manager displays the SQL Server 2005 node name instead of the virtual SQL Server 2005 server name

> **Best Practice:** Microsoft recommends upgrading to SQL Server 2000 Service Pack 4 on all SQL Server 2000 instances that will share a cluster with SQL Server 2005 instances before installing SQL Server 2005 in a side-by-side configuration. This will ensure that your SQL Server 2000 instances will be at the latest revision before placing any version of SQL Server 2005 on the same hardware. Upgrading to SQL Server 2000 Service Pack 4 is a prerequisite for any instance that will eventually be upgraded to SQL Server 2005.

32-Bit Failover Clustering on 64-Bit Systems

With SQL Server 2005, you can create a 32-bit failover cluster under the 64-bit version of Windows by using Windows on Windows (WOW). Nonetheless, Microsoft recommends that the version of SQL Server should match the version of your operating system. If you plan on deploying on a 64-bit version of Windows, you should use a 64-bit version of SQL Server if possible.

> **Important**: WOW mode is supported only on the x64 platform, not the Itanium IA64.

The side-by-side installation of the following tools and services on a 64-bit computer is not supported:

- 32-bit SQL Server management tools
- SQL Server Integration Services (SSIS)
- SQL Server Notification Services

For more information, see Knowledge Base article 906892: <u>You may experience slow performance when you run 32-bit SQL Server tools on 64-bit operating systems</u>.

Reporting Services, Notification Services, and Integration Services in a Failover Cluster

SQL Server 2005 Reporting Services, Notification Services, and Integration Services are not cluster-aware; they are always installed as stand-alone components, but can use a clustered SQL Server database.

Despite not being a cluster-aware application like Analysis Services or SQL Server itself, there is a way to install Integration Services in a failover cluster. Consult the article, How to: Configure Integration Services on a Cluster, which describes the process. Clustering Integration Services can provide better availability, but you may want to investigate other methods of making Integration Services available as this method may not meet your needs.

Analysis Services in a Failover Cluster

The ability to cluster Analysis Services is a new feature in SQL Server 2005. Before SQL Server 2005, the only way to make Analysis Services more available was to either configure it as read-only in a Network Load Balancing cluster, or create it as part of a standard Windows server cluster as a generic resource. Now, the ability to cluster Analysis Services is integrated into the core of the product, and it is part of the standard install just like the Database Engine has been since version 6.5.

Setup is flexible to suit your availability needs. You can cluster the SQL Server 2005 Database Engine, Analysis Services, or both. If both are clustered, SQL Server and Analysis Services can share the same cluster group, or they can be installed to the same cluster but in their own individual cluster groups in the same Windows server cluster, with each having its own name, IP address, and disk resources. Plan a clustered Analysis Services as if the Database Engine was not in the picture to ensure that you take into account all aspects of an Analysis Services implementation, including performance aspects specific to a deployment of Analysis Services.

If you cluster just the SQL Server Database Engine, you can cluster Analysis Services at a different time (and vice versa) and still add it to the same cluster group with SQL Server 2005 if you want. You do not have to add Analysis Services from the start; this allows you to expand your capabilities as your needs for various features within SQL Server change over time.

> **Best Practice:** Microsoft recommends placing all Analysis Services clustered installations in a single group with its own disk and IP resources. If you place Analysis Services and the Database Engine in the same cluster group, they will affect one another in the event of a problem, and cause both to fail over, whereas if they are kept separate, they will not.

Failover Clustering and Microsoft Virtual Server

Deploying a SQL Server 2005 failover cluster under Microsoft Virtual Server is fully supported; however Microsoft Virtual PC is not supported for production use. With Microsoft Virtual Server 2005 R2, a 64-bit (x64) host operating system is supported, but only 32-bit (x86) versions of Windows are supported under Virtual Server 2005 as a guest operating system.. For the latest information, consult the relevant Knowledge Base articles and documentation for supportability when you deploy SQL Server 2005.

To see how to deploy a clustered Microsoft Virtual Server, consult the TechNet article, [Using Microsoft Virtual Server 2005 to Create and Configure a Two-Node Microsoft Windows Server 2003 Cluster](). There is also a TechNet webcast that describes clustering SQL Server 2005 on Microsoft Virtual Server. The information is found in Knowledge Base article 891798: [How to cluster Microsoft SQL Server 2005 by using Microsoft Virtual Server]().

All limitations of the Virtual Server platform apply to installations, including the limit of only one shared disk per SCSI bus. This means that the quorum and all SQL Server data/log files will be installed on the same disk if there is only a single SCSI bus, which is not an optimal configuration. It is always better to deploy clustering on real hardware.

Microsoft currently supports no virtualization products other than Microsoft's own Virtual Server for a production deployment of SQL Server 2005 failover clustering. Should you deploy using another virtualization product, consult your virtualization vendor for issues or problems that may occur by virtualizing a Windows-based cluster with their software. Consult Knowledge Base article 897615: [Support policy for Microsoft software running in non-Microsoft hardware virtualization software]().

> **Important:** Whether clustered or not, running SQL Server under Virtual Server means that you must take several performance considerations into account such as the individual performance needs of a particular SQL Server instance (memory, CPU, disk), and adding the appropriate overhead associated with implementing using Virtual Server. Microsoft recommends implementing SQL Server 2005 in a production environment on real hardware, not on a virtual machine.

Supported Hardware

All SQL Server failover clusters must be deployed on fully supported, Microsoft-certified cluster solutions, even down to specific drivers for things such as host bus adapter (HBA) cards, which allow you to connect to a shared disk subsystem. This means that the underlying Windows server clusters must be listed in the Windows Server Catalog under the Cluster Solutions category. For more information, see the [Windows Server Catalog Cluster Solutions]() website.

The hardware for the cluster itself must be a solution listed in the Windows Catalog [Hardware Compatibility List](), or for Windows 20000 Datacenter, the [Windows 2000 Server Resources page](). Links to both lists can be found at the [Windows Server Catalog]() website.

The listed solutions include a great amount of detail. These details are crucial, as the solution you implement cannot deviate in some cases from what is listed. Under any circumstances, do not configure the cluster with the latest drivers unless the manufacturer has documented it as certified for use in a cluster. Most manufacturers who provide hardware as part of a clustered solution maintain a document that lists which versions, cards, and so forth, that are approved for use in a Windows clustered environment. Work with your preferred vendors to ensure that the implementation is done properly.

For full details, see the Knowledge Base article 309395, [The Microsoft support policy for server clusters, the Hardware Compatibility List, and the Windows Server Catalog](). Also see Knowledge Base article 327518: [INF: The Microsoft Support Policy for a SQL Server Failover Cluster](), as well as [Hardware and Software Requirements for Installing SQL Server 2005]().

Number of Nodes

SQL Server 2005 failover clustering supports different numbers of total nodes depending on the version of SQL Server and the version of the operating system. The following table lists the number of nodes supported by the various editions and operating systems. SQL Server 2000 is listed for comparison purposes only. For the latest number of supported nodes, consult Knowledge Base article 288778: <ins>Maximum number of supported nodes in a cluster</ins>.

Table 1. Nodes supported on a SQL Server 2005 failover cluster for a given edition of SQL Server and Windows.

	SQL Server 2000 Enterprise Edition	SQL Server 2005 Standard Edition	SQL Server 2005 Enterprise Edition
Windows 2000 Advanced Server (with SP4)	2	2	2
Windows 2000 Datacenter Server (with SP4)	4	2*	4
Windows Server 2003 Enterprise Edition	4	2*	8
Windows Server 2003 Datacenter Edition	**4 (32-bit), 8 (64-bit)**	2*	8

*Although you can have up to eight nodes in a server cluster itself, SQL Server can use only two with this edition of SQL Server, and are specified during the installation process.

> **Important:** All nodes in the same cluster must be of the same type of hardware (either all 32-bit or all 64-bit) and configured in the same way. That means if you are implementing a WOW failover cluster with x64 hardware, all nodes participating in the instance of SQL Server being installed must be using WOW.

Memory Support

SQL Server 2005 failover clustering supports different amounts of total memory depending on the version of the operating system it is deployed on. The following table lists the total memory currently supported by the various editions and operating systems. Typically, SQL Server 2005 supports whatever the operating system maximums are, so if a successor to Windows Server 2003 is released in the lifecycle of SQL Server 2005, check to see what new maximums SQL Server may support. SQL Server 2000 is listed for comparison purposes only.

Table 2. Memory support on a SQL Server 2005 failover cluster for a given edition of SQL Server and Windows.

	SQL Server 2000 Enterprise Edition	SQL Server 2005 Standard Edition	SQL Server 2005 Enterprise Edition

Windows 2000 Advanced Server (with SP4)	8GB	8GB	**8GB**
Windows 2000 Datacenter Server (with SP4)	32GB	32GB	**32GB**
Windows Server 2003 Enterprise Edition	32GB (32-bit) 64GB (64-bit)	32GB (32-bit) 64GB (64-bit)	**32GB (32-bit) 64GB (64-bit)**
Windows Server 2003 Datacenter Edition	**64GB (32-bit) 512GB (64-bit)**	**64GB (32-bit) 512GB (64-bit)**	**64GB (32-bit) 512GB (64-bit)**

Processor Support

SQL Server 2005 failover clustering supports different numbers of processors depending on the version of the operating system it is deployed on. The following table lists the number of processors currently supported by the various editions and operating systems. Typically, SQL Server 2005 supports whatever the operating system maximums are, so if a successor to Windows Server 2003 is released in the lifecycle of SQL Server 2005, check to see what new maximums SQL Server may support. SQL Server 2000 is listed for comparison purposes only.

Table 3. Processors supported on a SQL Server 2005 failover cluster for a given edition of SQL Server and Windows.

|	SQL Server 2000 Enterprise Edition	SQL Server 2005 Standard Edition	SQL Server 2005 Enterprise Edition
Windows 2000 Advanced Server (with SP4)	4	4	**4**
Windows 2000 Datacenter Server (with SP4)	8	4	**8**
Windows Server 2003 Enterprise Edition	8	4	**8**
Windows Server 2003 Datacenter Edition	**64**	**4**	**64**

Security Improvements

Security for SQL Server 2005 failover clusters is more secure out of the box than SQL Server 2000 and some of the basic requirements have changed. Unlike SQL Server 2000, in SQL Server 2005, all service accounts that will administer SQL Server, SQL

Server Agent, Full-Text Search, or Analysis Services must be added to a global domain group which is then added to the local Administrators group.

For a full set of privileges that must be configured for each service you are deploying, see Setting Up Windows Service Accounts. For general best practices around security for clusters at the Windows level, see the white paper, Server Clusters: Security Best Practices for Windows 2000 and Windows Server 2003, as well as Knowledge Base article 269229: How to manually re-create the Cluster service account.

One of the major changes during installation of a SQL Server 2005 failover cluster is the dialog box in Figure 2, which asks for domain groups for the clustered services. SQL Server failover requires domain connectivity, and therefore by default domains are required for a clustered installation of SQL Server, so creating these groups within Active Directory should be straightforward for the Active Directory or network administrator tasked with that responsibility. Most likely it will not be the database administrator, so the DBA may need to request that the proper person or group set up these groups properly. As a best practice, we recommend that you create four individual groups: one each for SQL Server, SQL Server Agent, Full-Text Search, and Analysis Services (if required).
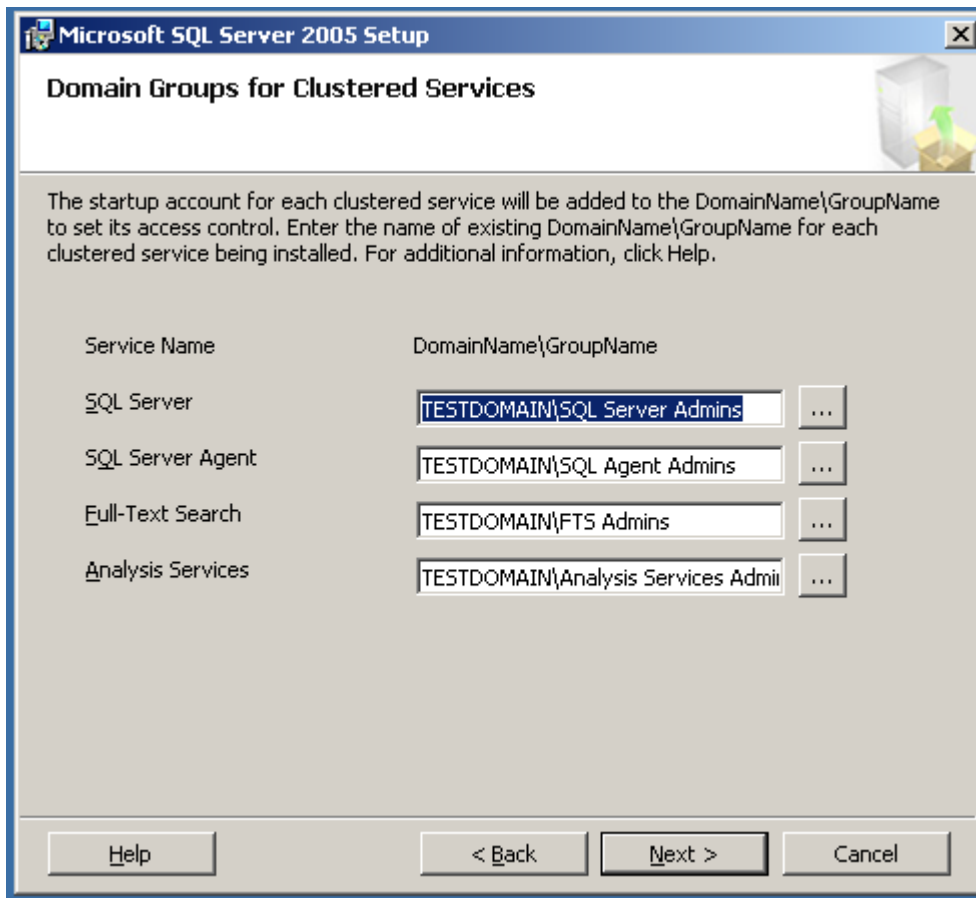
Figure 2. The SQL Server 2005 Installation Wizard allows you to add SQL Server services to startup accounts in different domains and Windows groups. The page will reflect the installation options you choose earlier in Setup.

These groups are configured at the domain level using the application Active Directory Users and Computers which is available on computers with access to edit Active

Directory, not on the individual cluster nodes. These steps may need to be performed by your Network Administrator.

> **Important:** All accounts and groups must be set up and verified before installing your cluster. If your server cluster is running SQL Server 2000 and SQL Server 2005 in a side-by-side configuration, never use the same service accounts for your SQL Server 2005 installation that you use for your SQL Server 2000 instances. Failure to use a separate set of service accounts will result in unexpected security behavior between your side-by-side and SQL Server 2005-only server cluster installations

To Create a Group

1. In Control Panel, double-click **Administrative Tools**, and open **Active Directory Users and Computers**.
2. In the left pane, expand the domain that will be used.
3. Right-click **Users**, point to **New**, and then click **Group**.
4. In the **New Object – Group** dialog, enter an appropriate name in the Group name text box. Select a group scope of Global. An example is shown in Figure 3.
5. Click **OK**.
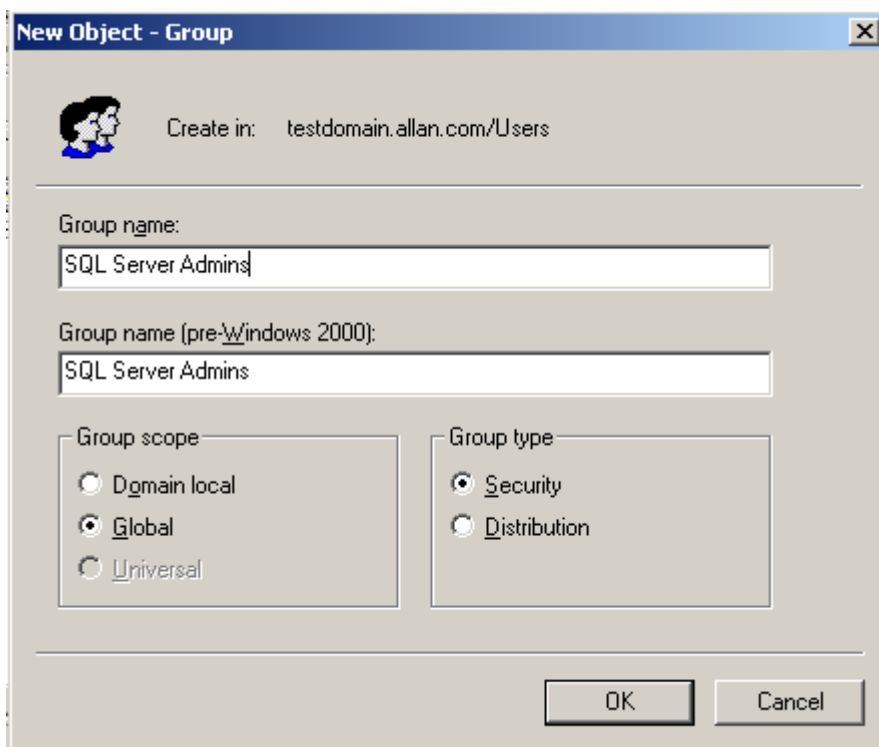
Figure 3. Creating new domain groups at the domain level using Active Directory Users and Computers.

To Add Users to a Group

1. Select the group which was just created, and double-click.
2. Click the **Members** tab.
3. Click **Add**.

4. On the **Select Users, Contacts, or Computers** dialog box, enter the names to be added in the lower half, and click check names, or navigate via the top half, select the object and click Add. When done, click **OK**.

5. The **Members** tab will now reflect the additions. An example is shown in Figure 4.

6. Click **OK** to close the dialog.
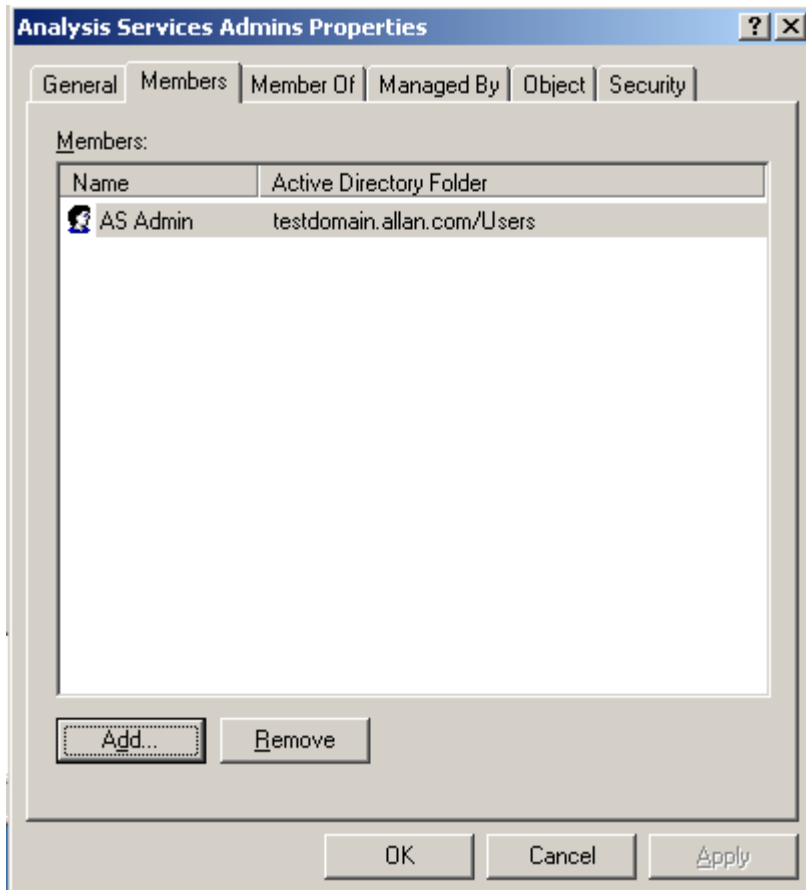
Figure 4. Add users to domain groups before installing a SQL Server 2005 failover cluster

For more information, see Domain Groups for Clustered Services.

Note: You must add the SQL Server administrator account to the group that will administer Full-Text Search. If you do not, you may encounter the following error during Setup:

"You do not have privileges to add accounts to the domain groups specified for this failover cluster. Ask your domain administrator for privileges to add new accounts to the domain groups, or log on using an account that does have permission.

"The domain group cannot be validated for the service Full-Text Search."

Disk Requirements

Each instance of SQL Server 2005 in a cluster must have at least one dedicated shared cluster disk. Disks cannot be shared among instances since they can only exist in a single cluster group, and only one SQL Server 2005 installation can be placed in a single cluster group.

In practice, this means that you are limited to a maximum of 22 or 23 usable, named disks with drive letters (assuming one local system drive, some sort of optical drive such as a DVD-ROM, and the possibility of a shared quorum drive). Unlike SQL Server 2000 failover clustering, SQL Server 2005 failover clustering does support the use of mount points (or "mounted volumes") to ease disk expansion and administration. All disks must be formatted as Basic disks. Dynamic disks are not supported out of the box in a cluster installation; see Knowledge Base article 237853 Dynamic disk configuration unavailable for server cluster disk resources. Third-party tools may be employed to utilize dynamic disks with a cluster installation.

If you define more than one logical disk per physical logical unit number (LUN), that disk will appear as one drive to the cluster, not two. So unless you are placing Analysis Services and the relational engine in the same cluster group (which is not a recommended configuration), there is no benefit to carving out two logical disks on one LUN in a cluster, since that LUN will be in a single cluster group.

Mount points are fully supported in a clustered configuration of SQL Server 2005, but they require the use of a drive letter just like a standard disk resource. As with a standard cluster disk resource, mount points must be added to the resource group containing SQL Server, and added as a dependency.

Whether or not your storage area network (SAN) shares disks for creating LUNs, make sure that the SAN engineer properly zones and masks the LUNs so that only the cluster nodes can see the LUNs it needs to see, and no one else can see or access them.

If Analysis Services is clustered in the same group with the Database Engine, Setup will place any shared system files for both products on the default drive selected during Setup itself. After that, Microsoft recommends providing Analysis Services with its own disk resources not only for availability, but to ensure that Analysis Services has what it needs to perform properly.

> **Note:** SQL Server 2005 supports iSCSI, and adheres to any Windows iSCSI support requirements. However, SQL Server installations require a dedicated, not shared, adapter for iSCSI devices.

Number of SQL Server Instances per Cluster

With SQL Server 2005, the maximum number of supported instances that can be deployed on a Windows Server failover cluster in a clustered SQL Server configuration is 25 for Enterprise Edition and 16 for Standard Edition. These are different from the limitation on a stand-alone system, which is a tested limit of 50. With SQL Server 2000, the limit was 16.

Even though you technically can configure up to 25 instances, the maximum number of instances you can deploy depends upon the hardware resources available, including available drive letters. Therefore, 25 deployed instances would mean that there is only one local drive for Windows, you are using a Majority Node Set (local quorum) server cluster that does not use a drive letter for a quorum disk, and you are only assigning one cluster disk per instance.

Before installing and configuring a multiple instance cluster, think about why you may need multiple instances of SQL Server: Is it for consolidation? Mixed version levels for supporting multiple applications with specific requirements? Easier administration? Meeting individual service level agreements (SLAs)? Maximizing hardware usage? These

and other issues must be thought out at the planning stage because the appropriate hardware must meet the expected capacity at any given moment.

In a clustered configuration, however, you must also take into account the failover scenario when planning multiple instances: if in a worst case scenario you have multiple nodes fail in the cluster and are down to one node, can a single node run all of your instances given the capacity planning that was done? If not, then capacity was likely not given sufficient consideration.

Installing a SQL Server 2005 Failover Cluster

Before installing SQL Server SQL Server 2005 in a clustered configuration, the underlying Windows failover cluster must be installed and fully tested. This installation must also include the configuration of a clustered Microsoft Distributed Transaction Coordinator (MS DTC). Since SQL Server is built upon the Windows layer, if it is not installed and verified before the SQL Server installation, the SQL Server setup process will fail.

Install your SQL Server 2005 failover cluster in the following order:

1. Install and cluster Windows, and make sure that all relevant patches are applied as per the Knowledge Base articles, Recommended hotfixes for Windows 2000 Service Pack 4-based server clusters or Recommended hotfixes for Windows Server 2003-based server clusters. For step-by-step instructions on how to configure a Windows Server 2003 failover cluster, read the Guide to Creating and Configuring a Server Cluster Under Windows Server 2003.

2. Configure a clustered MS DTC. For information on installing MS DTC, consult either Knowledge Base article 301600, How to configure Microsoft Distributed Transaction Coordinator on a Windows Server 2003 cluster or Knowledge Base article 204209, How to rebuild or move a MSDTC installation to be used with a SQL failover cluster.

3. Verify that the underlying Windows failover cluster is properly configured and working.

4. Verify that there is at least one dedicated shared disk on the cluster for the SQL Server instance.

5. Verify that a dedicated and unique IP address and name (whether it will be a default or a named instance) has been reserved for SQL Server.

6. Create the users and groups at the domain level for use with your clustered instance(s) of SQL Server 2005.

7. Add the groups to your local nodes and give the groups the proper rights on each node as per the SQL Server 2005 Books Online topic Setting Up Windows Service Acccounts.

8. Cluster SQL Server 2005.

9. Perform any post-installation tasks.

Once the Windows and MS DTC portions of the installation are complete, the administrator must create a cluster group for the intended SQL Server installation, and that group must contain at least one shared cluster disk for use by SQL Server. Detailed requirements for installation are described in Installing SQL Server 2005 and Before Installing Failover Clustering.

> **Note:** Before installation, all settings and policies should be identical across all nodes of the cluster for consistent behavior, including all security-related policies, hotfix installations, driver installations, and locales. If a change is made after installation, that change must be reflected on all nodes as well.

Before installation, we recommend logging off all users on the cluster nodes except the node where setup is being started.

> **Best Practice:** SQL Server 2005 requires the .NET Framework version 2.0 on all nodes. While Setup will do install .NET Framework 2.0 on its own, we recommend installing this prerequisite before attempting to install SQL Server 2005 to speed up the installation process.

SQL Server 2005 failover clustering can be installed in one of three ways:

- Running Setup.
- From the command prompt with switches.
- From the command prompt with a fully configured .ini file.

For more information about using Setup, see Appendix A as well as How to: Create a New SQL Server 2005 Failover Cluster (Setup). For more information about installing from the command prompt, see How to: Install SQL Server 2005 from the Command Prompt. A template for the .ini file can be found on the installation media under \Servers with the name template.ini. Examples can also be found in Appendix C.

Installation from the command prompt is an enhancement to SQL Server 2005. Other enhancements to the Setup process include the following:

- The current node's name is written at startup to the SQL Server log and to the Windows Event log.
- Installation proceeds in parallel on all nodes.
- The installation rolls back on all nodes if Setup fails on any one node.
- Logging of Setup operations is improved.
- Error messages during Setup are improved.
- Setup Consistency Checker detects the state of machines before installation.
- Setup now includes error reporting.
- Unattended setup has feature maintenance, allowing you to later add, remove, or reconfigure your SQL Server installation.
- The Installation Wizard includes progress reporting, and Setup can be canceled.

Installing on a Domain Controller

Because of the performance requirements of a SQL Server 2005 failover cluster, clustered installations are not supported on a node that is a domain controller. If you install on a domain controller and encounter a problem, you may be required to reconfigure the cluster. Here are some of the problems associated with installing on a domain controller:

- Overhead associated with running a domain controller. A domain controller that is otherwise idle can use anywhere between 130 to 140 MB of RAM, which includes running Windows Clustering. There is also replication traffic if the domain controller has to replicate with another domain controller within the domain and across domains.

- The first domain controller in the forest takes on all flexible single master operation roles (see Knowledge Base article 197132: <u>Windows 2000 Active Directory FSMO Roles</u>). You can redistribute these roles to each node. However, if a node fails over, the flexible single master operation roles that the node has taken will no longer be available. You can use Ntdsutil to forcibly take away the roles and assign them to the node that is still running (see Knowledge Base article 223787: <u>Flexible Single Master Operation Transfer and Seizure Process</u>). For information about placement of flexible single master operation roles throughout the domain, see Knowledge Base article 223346: <u>FSMO Placement and Optimization on Windows 2000 Domain Controllers</u>.

- Performance demands of a domain controller may prevent the cluster service from accessing the quorum drive, which may cause inadvertent failover to a secondary node.

- If you are using a failover cluster where the nodes are also serving as domain controllers, you can encounter problems since the nodes would be dependent on each other for things like primary and secondary Active Directory and DNS services. This means that if one node goes down, you may have the potential of causing larger outages of availability to your clustered services.

For more information about installation on a domain controller, see Knowledge Base article 281662: <u>Windows 2000 and Windows Server 2003 Cluster Nodes as Domain Controllers</u>.

Troubleshooting Installation Problems

If installation fails, check not only the Windows Event Logs, but also the logs that are 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 most likely 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 <u>Failover Cluster Troubleshooting</u>.

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\NtfsDisable8dot3Name Creation

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.

Post-Installation Tasks

After installing your SQL Server 2005 failover clustering instance, you may need to perform some additional tasks to make your instance fully usable.

Install SQL Server Tools on Additional Nodes

SQL Server 2005 Setup no longer installs the tools on any of the nodes other than the node that initiated Setup. If tools are needed on other nodes, they must be installed manually after basic setup. This can be done using either Setup or the command prompt. For instructions on installing the tools via Setup, consult SQL Server Books Online, or to do it using the command prompt or an .ini file, see Appendix C.

Assign a Manual IP Port

Like SQL Server 2000, SQL Server 2005 is configured by default to use a dynamic port number that will be assigned when SQL Server resources are started. If you want to change this to a static port to ensure that the port number stays the same in a failover or for security reasons where you would not want a default port number such as 1433, follow these steps:

1. Start SQL Server Configuration Manager.

2. In the left pane, expand **SQL Server 2005 Network Configuration**, and select **Protocols for** *<INSTANCE>*, where *<INSTANCE>* is your instance name.

3. In the right pane, double click **TCP/IP**.

4. In the **TCP/IP Properties** dialog box, click the **IP Addresses** tab.

5. Scroll to the bottom, and look at the **IPAll** section. If there is a value for **TCP Dynamic Ports**, delete it. Select the **TCP Port** option, enter a valid port number, and click **OK**.

6. Click **OK** at the warning.

7. Restart the SQL Server service to enable the change.

Add Additional Cluster Disks

During Setup, only one cluster disk can be added. If other disks are required, they must be added after the basic failover clustering installation per resource. That means that for both Analysis Services and for SQL Server, the disks must be added. For instructions, see How to: Add Dependencies to a SQL Server 2005 Resource.

> **Note:** Adding a disk as a dependency will incur downtime, so you should plan your disk configuration at the time of initial install.

If you want to use mount points with SQL Server 2005, follow the instructions found in the Knowledge Base article 280297: <u>How to configure Volume Mount Points on a clustered server</u>.

> **Important:** If you are using mount points, you must place the mount point disk resource in the proper cluster group with SQL Server or Analysis Services and make it a dependency of its host drive.

Change the Affect the Group Property of the SQL Server or Analysis Services Resource

On the **Advanced** tab of the SQL Server or Analysis Services resource, the **Affect the group** checkbox is not checked by default.This means that if the SQL Server or Analysis Services resource fails, the resource group will not automatically fail over to another node. This is a change in behavior from SQL Server 2000 failover clustering. This behavior was changed because you can put both Analysis Services and SQL Server in the same resource group, and you may not want to have one affect the other. However, under normal circumstances, you do want SQL Server to automatically fail over to another node. Check **Affect the group**.

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.

Upgrading to SQL Server 2005 Failover Clustering

In SQL Server 2005, you can no longer upgrade a stand-alone instance of SQL Server to a clustered instance via Setup. Only a clustered instance of SQL Server can be upgraded to a SQL Server 2005 failover cluster.

This means that if you want to convert a stand-alone instance of SQL Server to a failover cluster instance, whether it is an existing SQL Server 2000 instance, or a newer instance of SQL Server 2005, you must first install a clustered instance of SQL Server 2005 and then migrate your databases, logins, jobs, and so on using your preferred

method. The Copy Database Wizard is not supported in SQL Server 2005, but there are a few other options that can achieve the database migration from a stand-alone server:

- Detach and attach the databases.
- Use log shipping.
- Use backup and restore.

   Note: If you install a side-by-side failover cluster, you cannot reuse the existing SQL Server name used by the previous version (default or named instance) until you disable or uninstall the already existing SQL Server instance with the name to be used. The same goes for a stand-alone instance that is now a failover cluster, except the entire server will have to be shut down because a stand-alone instance assumes as part of its name the underlying server. This must be taken into account during planning for the upgrade.

If you plan to use log shipping as your upgrade method, you must deploy your own custom version. The built-in log shipping functionality of SQL Server 2000 and SQL Server 2005 do not work together. You can manually create your own log shipping since the restore a database or transaction log backup created with SQL Server 2000 under SQL Server 2005, and SQL Server 2005 will perform the necessary upgrades in the restore process.

For full instructions on doing an in-place upgrade using Setup, see How to: Upgrade to a SQL Server 2005 Failover Cluster (Setup).

For full details on the upgrade process for SQL Server 2005, see Upgrading to SQL Server 2005 and How to: Upgrade to a SQL Server 2005 Failover Cluster (Setup). For detailed information on upgrading to SQL Server 2005 including cluster scenarios, read the SQL Server 2005 Upgrade Technical Reference Guide.

Administering a SQL Server 2005 Failover Cluster

This section explains how to administer your SQL Server 2005 failover clustering installations.

   Note: SQL Server 2005 tools are 32-bit, and run under Microsoft Windows® on Windows (WOW) on 64-bit platforms. For more information on how they behave under Microsoft's 64-bit platforms, see the Knowledge Base article 906892: You may experience slow performance when you run 32-bit SQL Server tools on 64-bit operating systems.

SQL Writer and Failover Clustering

The SQL Writer Service provides functionality for backup and restore of SQL Server 2005 through the Volume Shadow Copy Service (VSS) framework.

The SQL Writer Service is installed on each node of the cluster during Setup, but is disabled by default. Enable it if necessary.

SQL Server Browser Service

The SQL Server Browser Service directs connection attempts that use instance names to the proper instance on the server. It is installed on each node in the cluster and is

not a clusterable resource. The SQL Server Browser Service is started automatically for a clustered instance of SQL Server.

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.

SQL Server 2005 Surface Area Configuration

The SQL Server 2005 Surface Area Configuration tool is a new security tool. It allows administrators to easily manage services and features of SQL Server 2005. For more information, see [SQL Server Surface Area Configuration](#).

The tool is cluster-aware, but it must be configured properly to administer a clustered instance of SQL Server 2005. By default, it only tries to connect to a local instance of SQL Server. To configure Surface Area Configuration to work with a clustered instance, perform the following steps:

1. Start the tool from **Start→Programs→Microsoft SQL Server 2005→Configuration Tools→SQL Server Surface Area Configuration**.

   When the main screen of SQL Server 2005 Surface Area Configuration is displayed, it defaults to localhost. If you try to use the tool now, you will see the following error:

   "You cannot configure surface area of clustered services by connecting to a computer name. Connect to the virtual server to configure clustered services."

2. Click the **change computer** link.

3. In the **Select Computer** dialog box, click **Remote computer** and enter the failover clustering instance name used during the installation process. If it is a named instance, do not enter the full name, just the failover clustering instance.

4. To verify the failover clustering instance name, start Cluster Administrator, select the cluster group with the SQL Server 2005 resources, and look next to the SQL Network Name resource for the failover clustering instance name. An example is shown below.
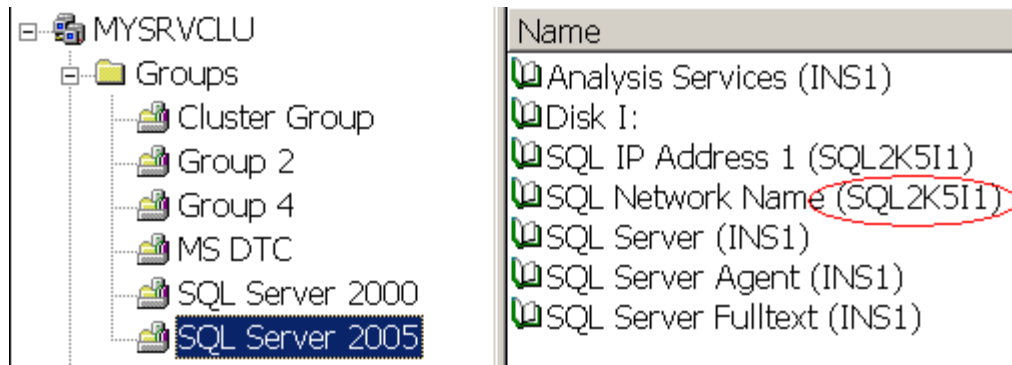
Figure 5. SQL Network Name resource

The main screen will now reflect the name change.

5. To administer SQL Server, click either **Surface Area Configuration for Services and Connections**, or **Surface Area Configuration for Features**. Note that when you select either option, the tool indicates that the instance that is currently being administered is clustered.

    Note: SQL Server 2005 Surface Area Configuration can only work with one instance at a time. If you have more than one failover clustering instance, repeat the steps above for the other instances.

SQL Native Client

Failover clustering in SQL Server 2005 no longer depends on a specific version of MDAC, unlike previous versions of SQL Server. SQL Server 2005 uses SQL Native Client for connectivity. This means that when installing or upgrading SQL Server, or installing a SQL Server service pack, MDAC will not be bundled with the installer and will use the version of MDAC that is currently installed on the host server.

   Important: If you have a side-by-side SQL Server 2000 and SQL Server 2005 failover cluster configuration, SQL Native Client will become the connectivity layer for the SQL Server 2000 instances as well. At a lower level, the SQL Server 2000 instances will actually be using the SQL Server 2005 resource DLL to monitor the instance's health.

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.

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.

SQL Server Surface Area Configuration

SQL Server Surface Area Configuration can also be used to stop and start the SQL Server-related services in a clustered configuration. To stop or start a SQL Server 2005 failover clustering instance using SQL Server Surface Area Configuration, follow these steps:

1. Start SQL Server Surface Area Configuration.

2. Click **Surface Area for Services and Connections**.

3. In the left pane, click the service that you want to start or stop, and on the right side, click the appropriate action.

Cluster Administrator

Cluster Administrator can be used to stop and start the SQL Server-related services in a clustered configuration. To stop or start a SQL Server 2005 failover clustering instance using Cluster Administrator, follow these steps:

1. Start Cluster Administrator.

2. In the left-hand pane, expand **Groups**, and select the cluster group which contains the SQL Server 2005 instance you would like to start or stop.

3. In the right-hand pane, right click the SQL Server resource, and click **Take Offline** to stop the instance or **Bring Online** to start the instance.

   Note: To use this method you must have sufficient administrative rights to run Cluster Administrator and connect to the server cluster.

Viewing the Properties of a Failover Clustering Instance

Viewing properties or status of various aspects of your failover cluster in SQL Server 2005 has changed from SQL Server 2000. The following sections describe the methods available.

Drive Configuration

SQL Server 2005 introduces the dynamic management view (DMV) `sys.dm_io_cluster_shared_drives`, which has the same functionality as the function `fn_servershareddrives` in SQL Server 2000. This DMV returns all of the disk drives that are part of the resource group and can be used to hold data and log for this instance of SQL Server. The user must have the VIEW SERVER STATE permission to access this DMV.

To execute, run `SELECT * FROM sys.dm_io_cluster_shared_drives` in a query window. For more information see sys.dm_io_cluster_shared_drives.

> **Note:** `fn_servershareddrives` is still available and supported in SQL Server 2005; Microsoft recommends changing code that references it to use the new DMV. To access `fn_servershareddrives` in SQL Server 2005, the user must have the VIEW SERVERSTATE permission.

IP Address

SQL Server Configuration Manager can display the properties of the TCP/IP protocol. However, it does not display the clustered IP address or addresses that were 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, perform the following steps:

1. Start Cluster Administrator.
2. In the left pane, expand **Groups** and select the cluster group that contains the desired SQL Server 2005 instance.
3. 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.
4. 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.

> **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. For more information, see SERVERPROPERTY.

Online Status

For information on tools and processes for verifying whether SQL Server is running, see "Starting, Stopping, and Pausing Services" earlier in this white paper.

Adding or Removing a Node

There are cases where you may need to remove, or evict, a node from the cluster. There are two levels of removal when it comes to SQL Server: removing it from the definition 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.

Using the Command Prompt

See Appendix C.

Renaming a Clustered Instance of SQL Server

SQL Server 2005 supports renaming an instance after it is installed, unlike SQL Server 2000, which required a full uninstall and then a reinstall to rename the failover clustering instance. The renaming process uses Cluster Administrator. For instructions, see How to: Rename a SQL Server 2005 Virtual Server.

> **Note:** Renaming a clustered will require a small amount of downtime and the proper privileges at the Windows level.

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.

> **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](#).
- Remove it via Setup as detailed in [How to: Remove a SQL Server 2005 Failover Clustered Instance (Setup)](#).

If there are problems in the uninstall process, see [How to: Manually Uninstall a SQL Server 2005 Failover Cluster](#).

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.

Performance

See [Optimizing Failover Cluster Performance](#) for information about further optimizing your cluster installations. This topic focuses on the following issues:

- Write caching
- Use of file shares

- Maximum and minimum memory size
- Processor affinity
- Use of fixed memory settings

Other resources for optimizing your SQL Server 2005 installations are available, including Optimizing Server Performance. Database Performance covers a wide range of considerations and also has a dedicated section on hardware considerations, including Comparing Different Implementations of RAID Levels.

Additional white papers about performance and optimization of SQL Server 2005 will also be released, so always check the SQL Server TechCenter for the latest documents.

Conclusion

Failover clustering remains one of the core SQL Server high-availability technologies. It is enhanced in SQL Server 2005, and can be combined with other SQL Server high-availability technologies to create even greater availability for your instances and databases. Implementing a SQL Server 2005 failover cluster requires a solid understanding of the underlying technology and an appropriate level of planning. A properly configured failover clustering implementation, along with good people, processes, and planning, will provide increased availability for your SQL Server 2005 implementations.

Appendix A – Step-by-Step Instructions for Installing or Upgrading Failover Clustering (Database Engine or Analysis Services)

Follow the instructions below to install a clustered instance of SQL Server 2005 using SQL Server Setup.

1. Log into a node of the Windows server cluster as a domain user with administrative privileges on all nodes of the cluster, such as the user Cluster Administrator.

2. Insert the SQL Server 2005 installation media into the node which you logged into, or navigate to the proper network location at your business. If you have autorun enabled on your CD or DVD drive, the following menu below should appear. If the screen does not appear, navigate to the Servers directory of the media and run setup.exe. Click **Server components, tools, Books Online, and samples**.
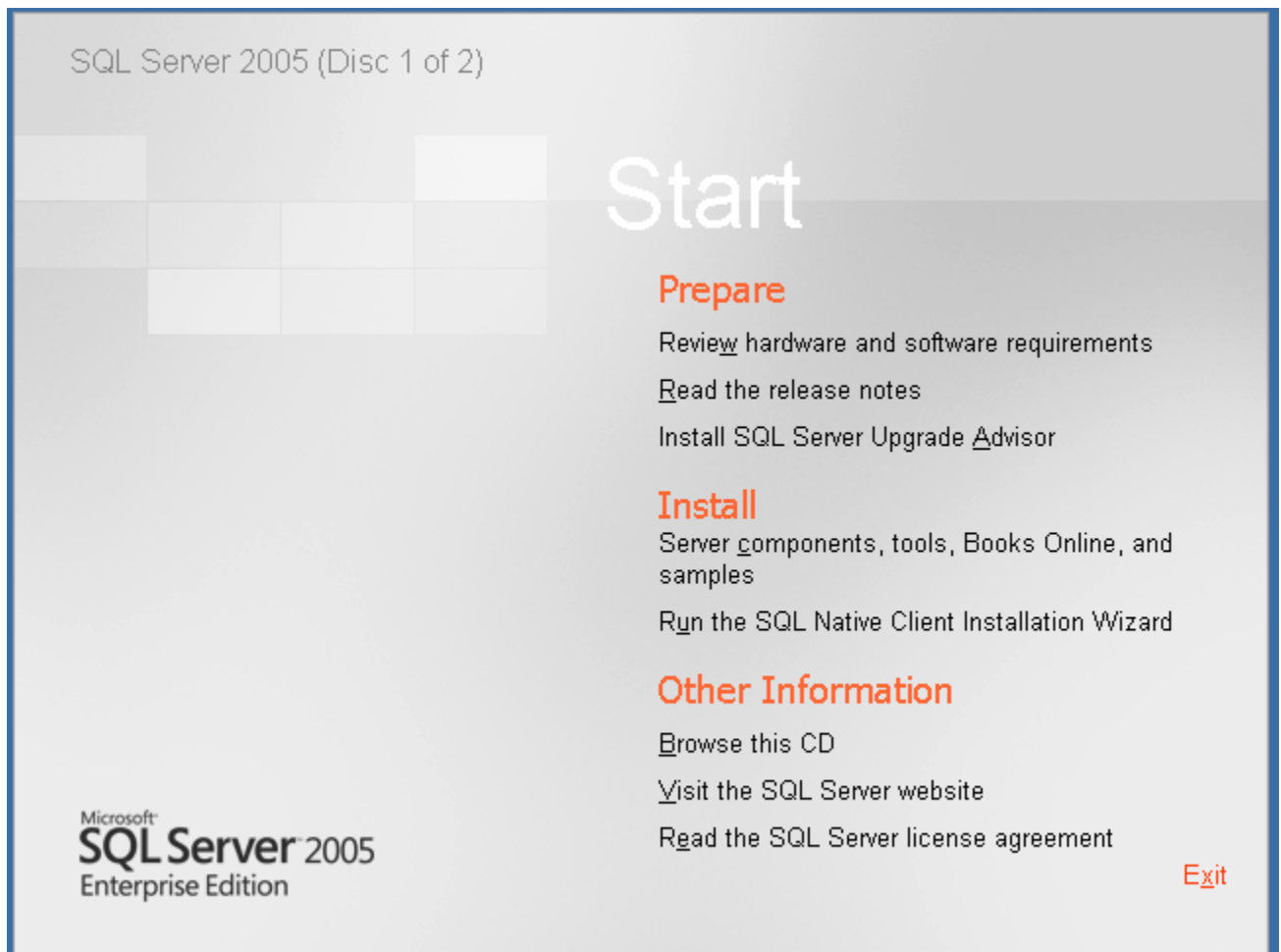
Figure 6. SQL Server 2005 Setup Start page

3. Read the End User License Agreement (EULA). Select **I accept the licensing terms and conditions** and click **Next**.
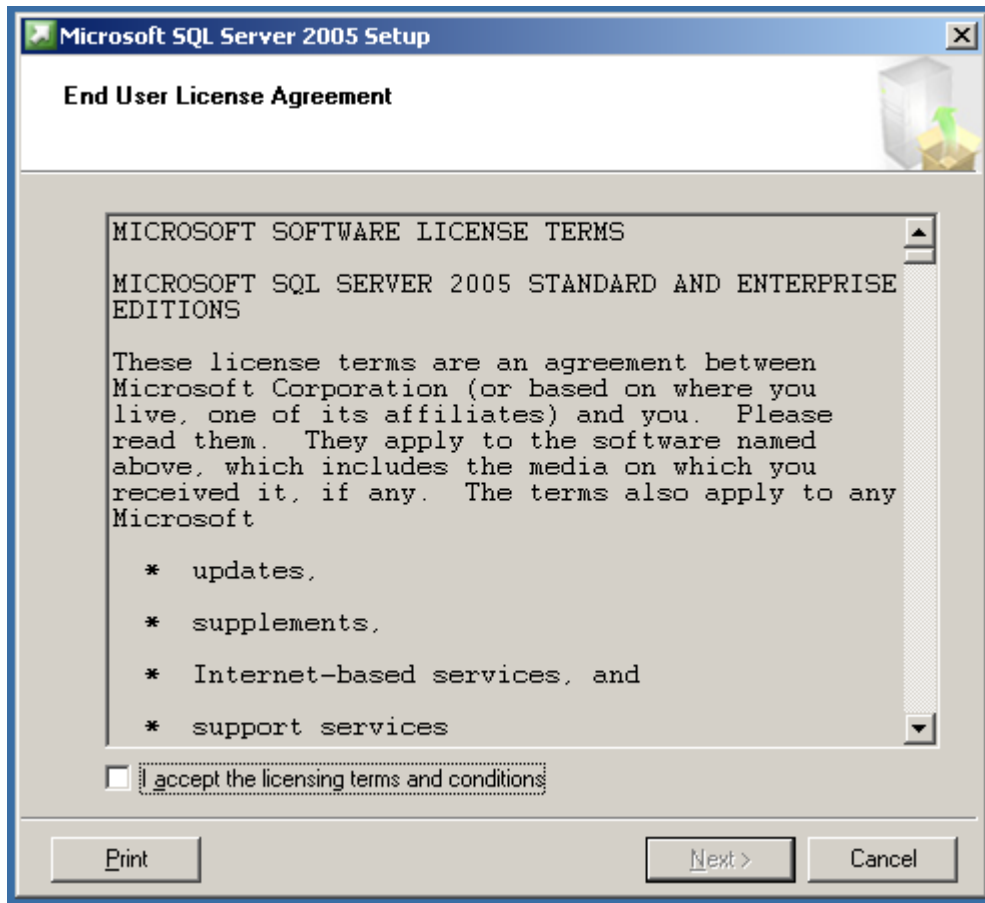
Figure 7. End User License Agreement

4. Setup will now install any prerequisites on the node which started the install process. During the process, a green checkmark will appear next to completed tasks and a red arrow pointing right will indicate the current task being executed. When all tasks are completed, click **Next** to continue.

   TIP: To speed up the initial phase of the installation or upgrade process, we recommend installing the proper version of the .NET Framework (which can be downloaded from the Microsoft .NET Framework Developer Center) on each node before starting the SQL Server installation so that SQL Server Setup will not have to do it.
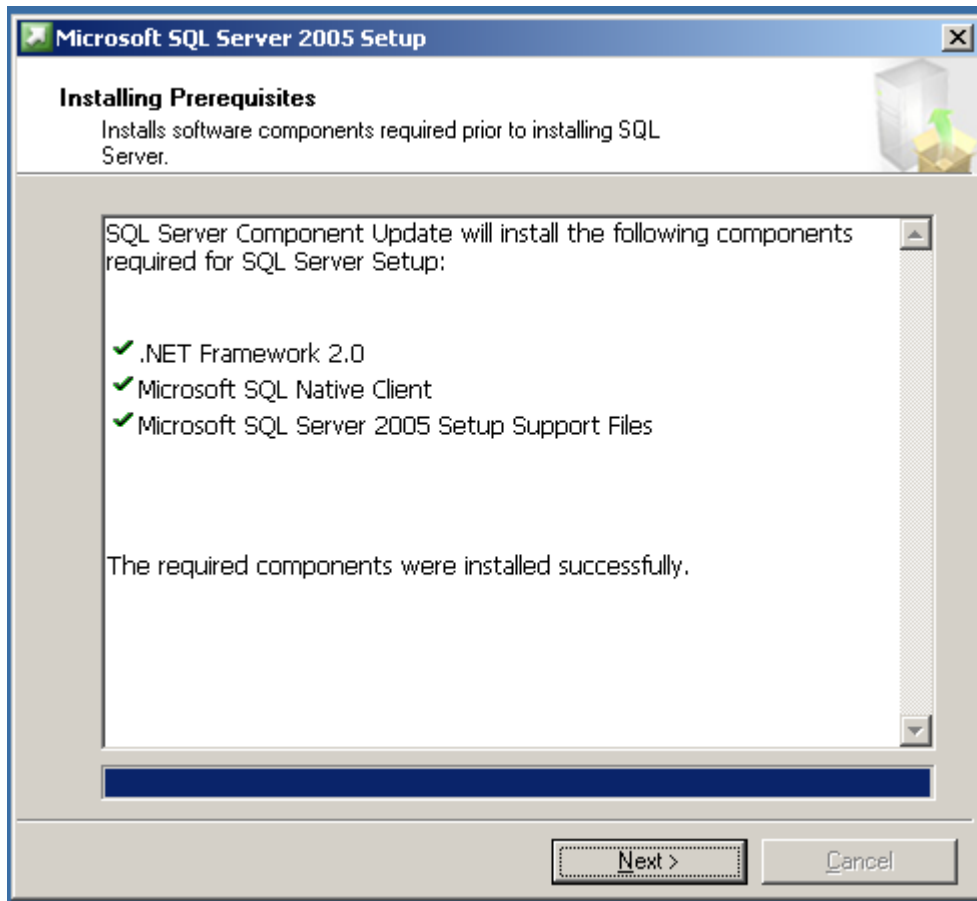
Figure 8. Installing Prerequisites page

Setup will now perform a configuration check of the node that started the installation process.
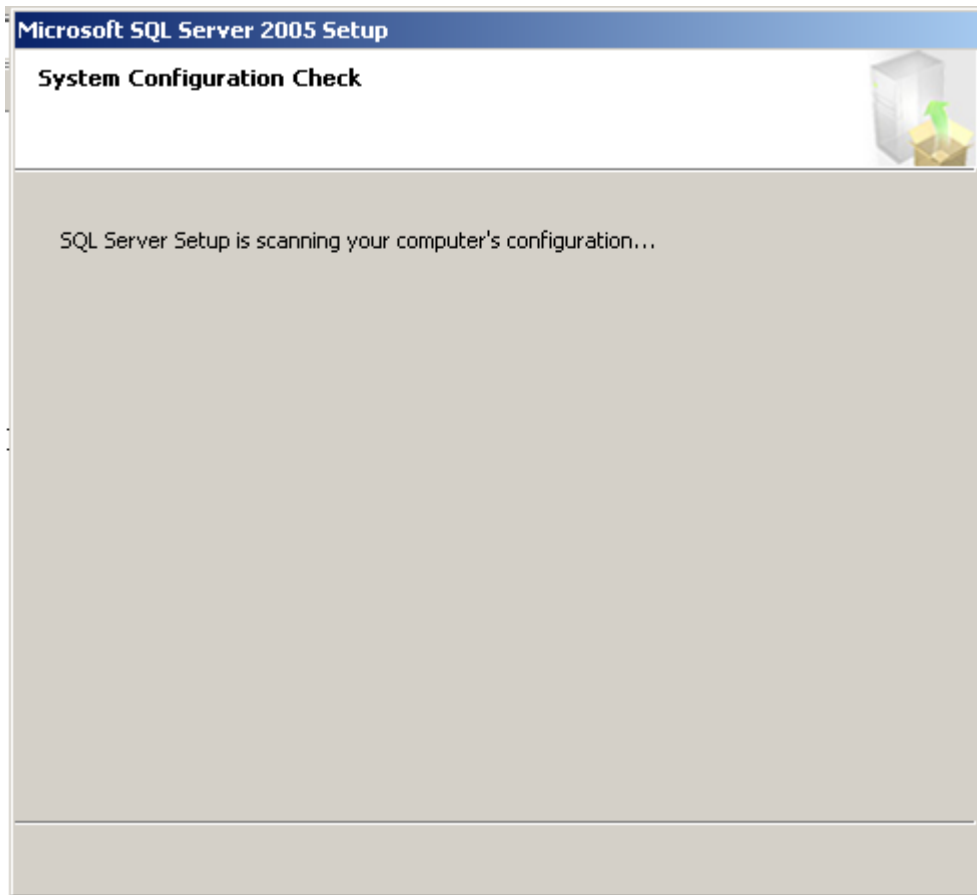
Figure 9. System Configuration Check page

5. When the Welcome to the Microsoft SQL Server Installation Wizard page appears, click **Next**.

Figure 10. Welcome to the Microsoft SQL Server Installation Wizard page

6. Setup will now do a more exhaustive check of the node which started the installation process. Any warnings or errors will be detected, and they can be seen by clicking **Messages** next to the appropriate item, or by viewing a report via the **Report** dropdown. The report can be viewed on the screen, saved to a file, copied to the Clipboard, or e-mailed.

> **Note:** If you have any errors, fix them before proceeding with the installation as they may either cause the installation to fail or compromise the stability of the installation. Exit out of Setup and rerun when complete. Warnings are not fatal, but they should be evaluated to see if they are valid and need to be fixed.
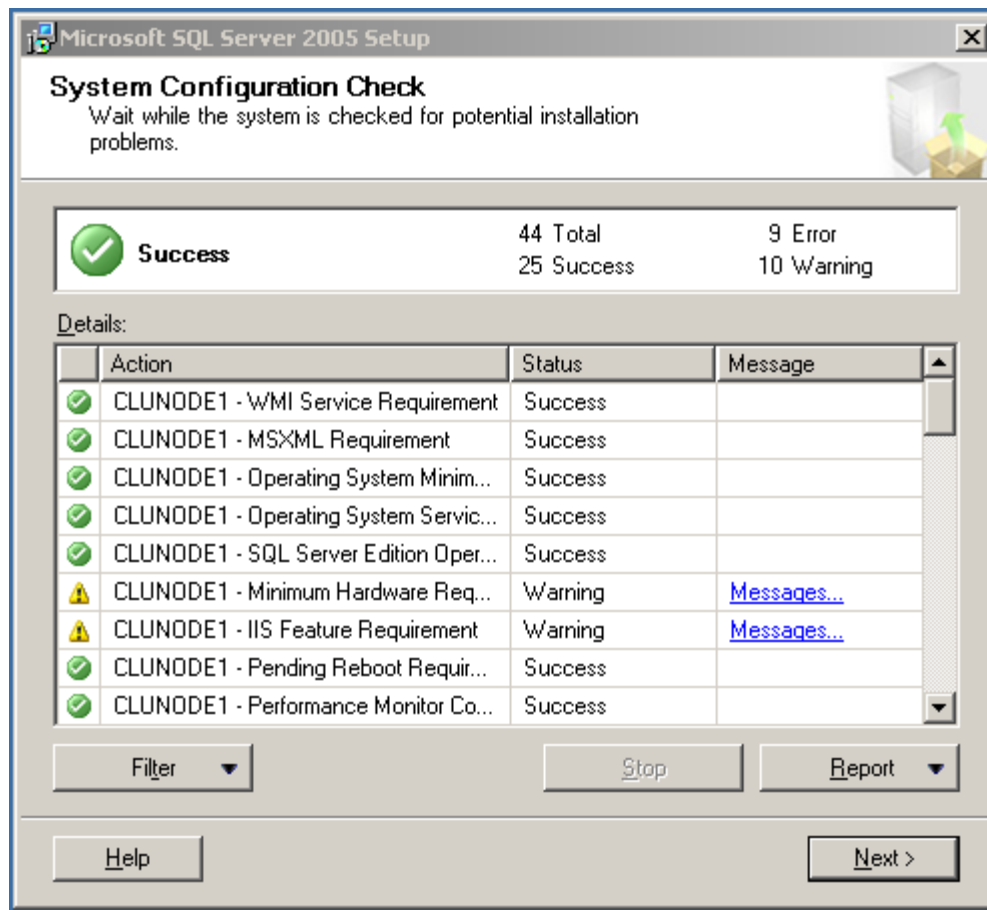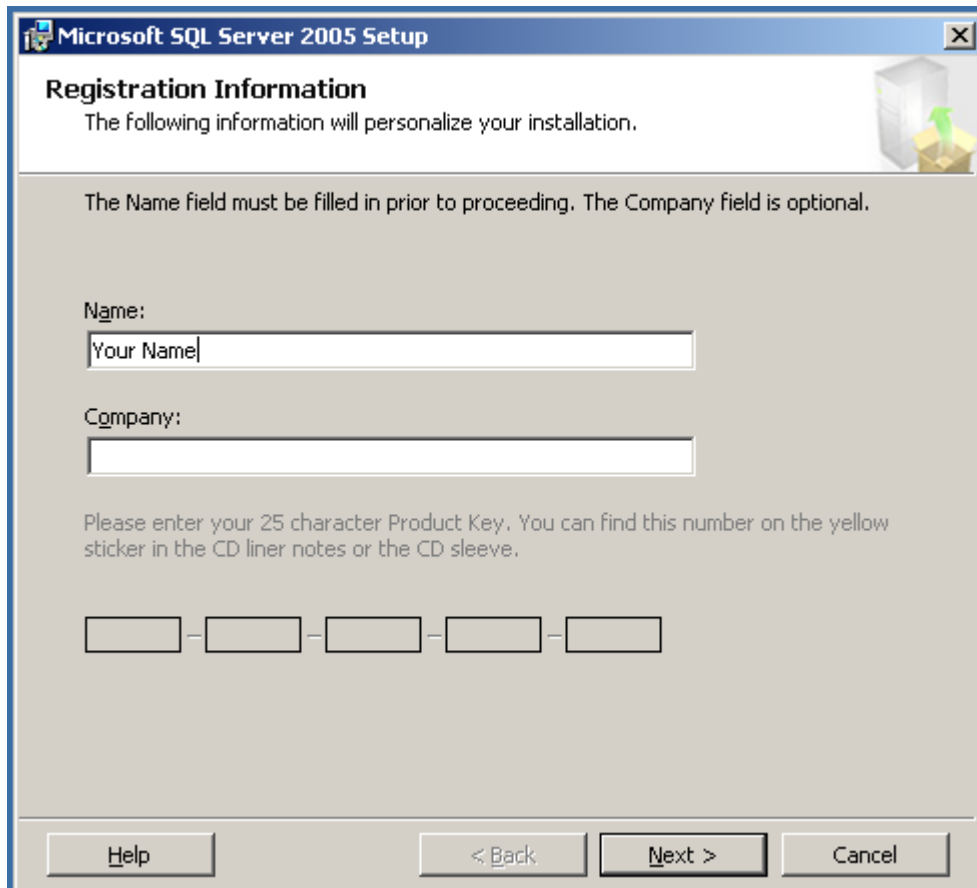
Figure 11. System Configuration Check

7. When the check is complete, click **Next**. Setup will prepare the installation.
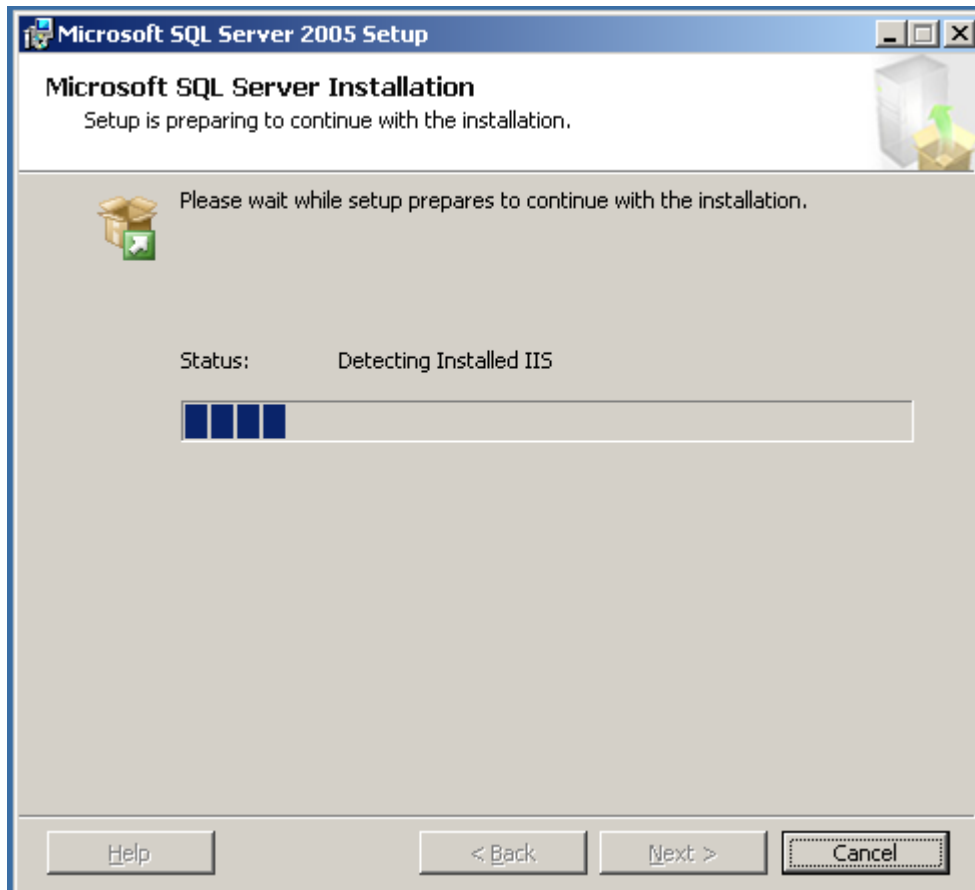
Figure 12. Microsoft SQL Server Installation page

8. Enter your name and a valid Product Key (PKD) from your installation media. Click **Next**.

Figure 13. Registration Information page

9. Select the components that you want to install during the Setup process. You do not need to select all features, as they can be added later. To install a relational engine failover clustering instance, select the **SQL Server Database Services** check box and then select the **Create a SQL Server failover cluster** box. If you do not select the latter, it will be installed as a stand-alone instance. To install an Analysis Services failover clustering instance, select the **Analysis Services** check box and then select the **Create an Analysis Services failover cluster** check box. If you do not select the latter, it will be installed as a stand-alone instance. Click **Next** when done.
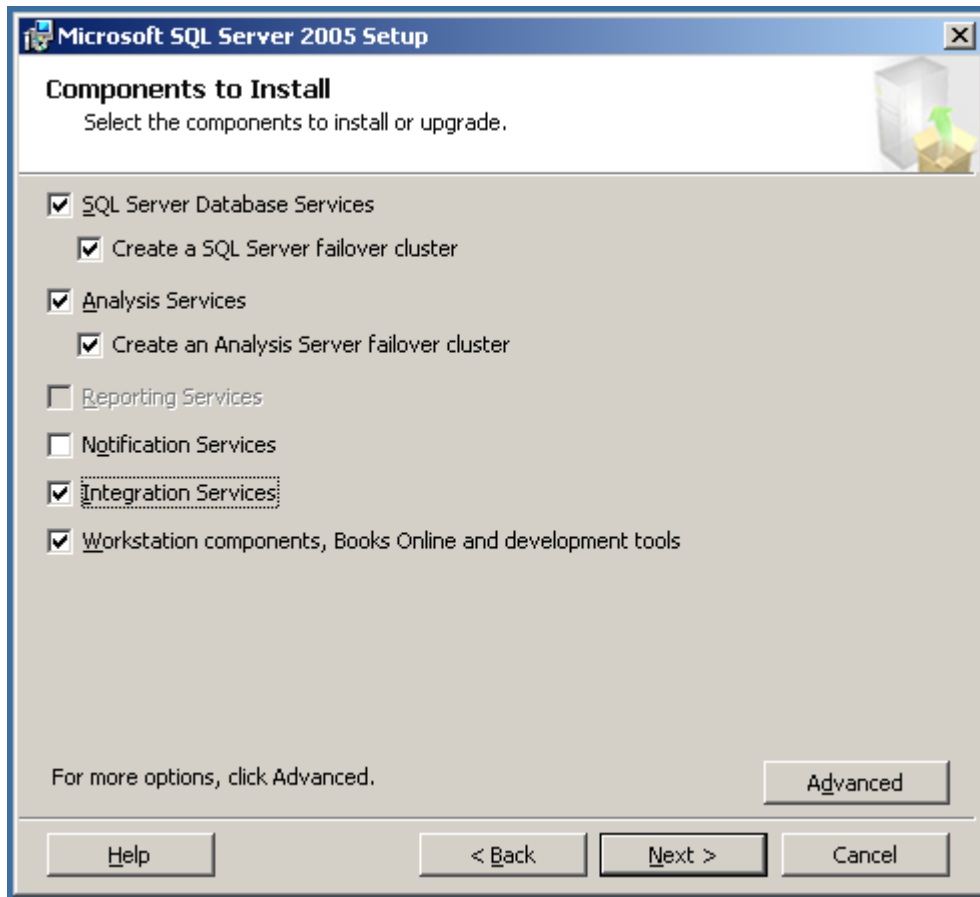
Figure 14. Components to Install page

If you click **Advanced**, the Feature Selection page will be displayed. Here you can further customize your SQL Server installation; however it is recommended that you not change any of the options unless you are familiar with the workings of SQL Server 2005. These screens are described here so you can see what is done on them should you choose the Advanced option.
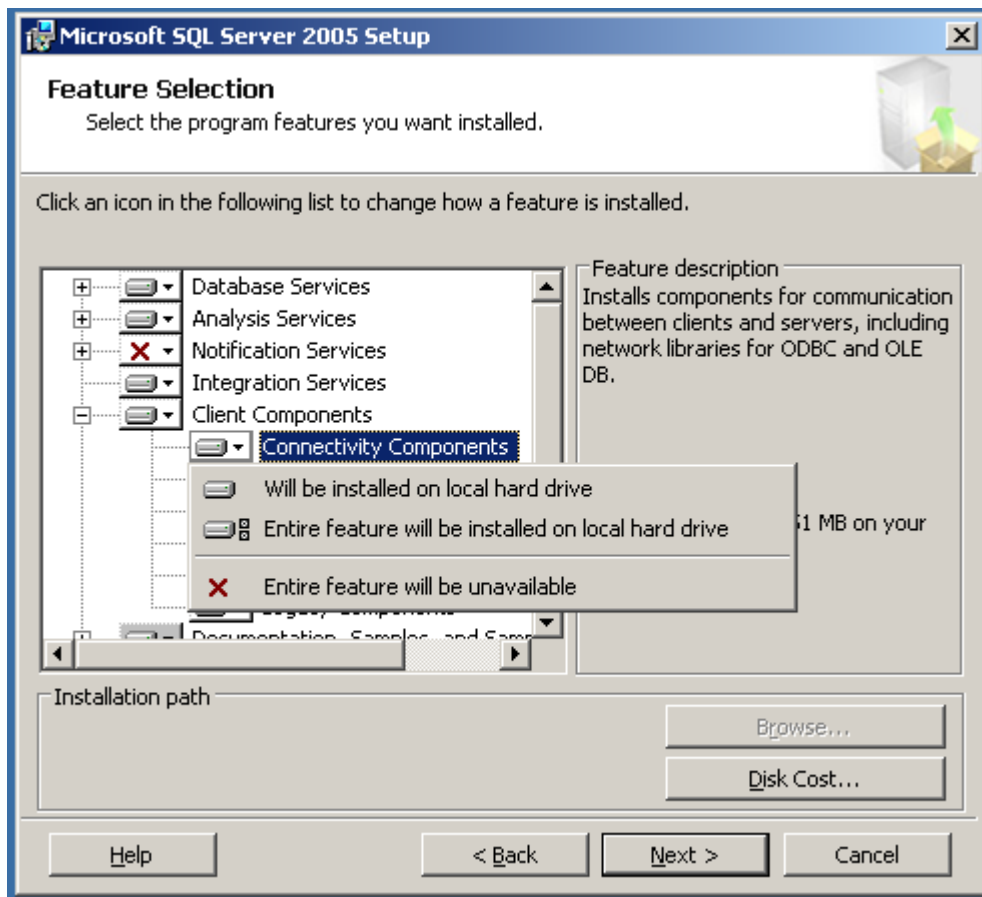
Figure 15. Feature Selection page

If you click **Browse**, the Change Folders page appears. On this screen you can change the top-level installation folder where the SQL Server program files will be installed on the nodes.
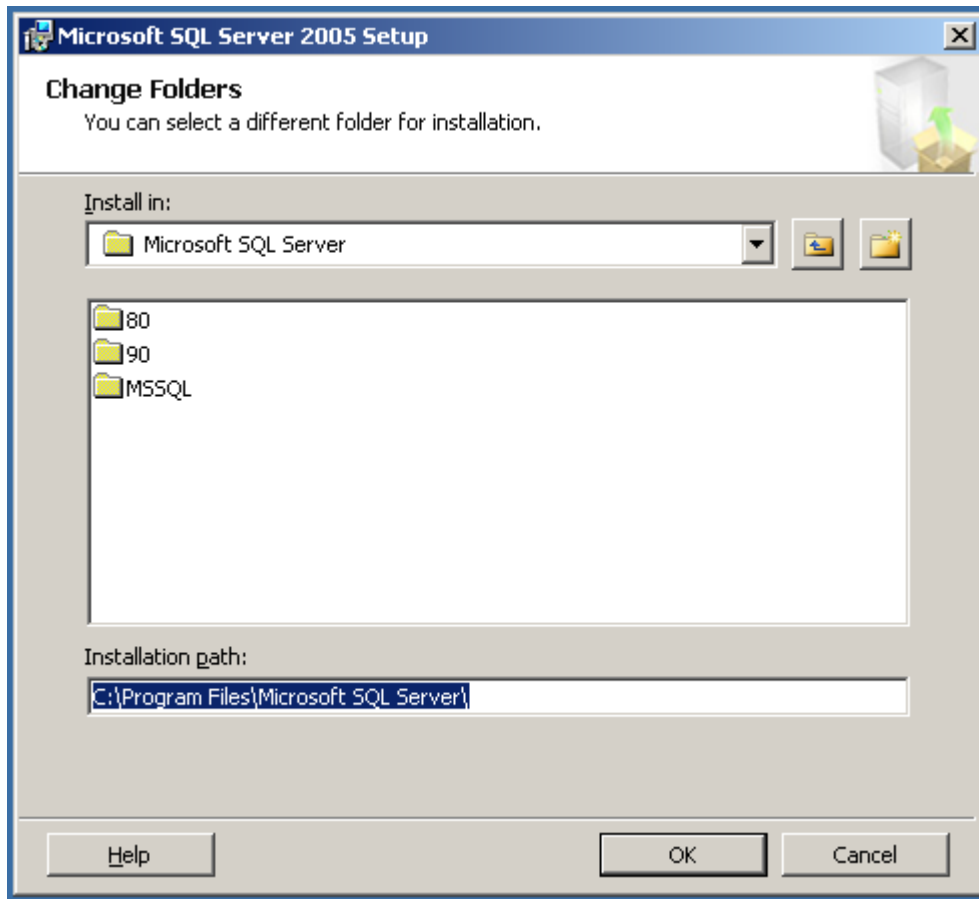
Figure 16. Change Folders page

If you click **Disk Cost**, Setup will display the disks available, how much space is left, and how much space is needed by SQL Server for the installation to complete successfully.
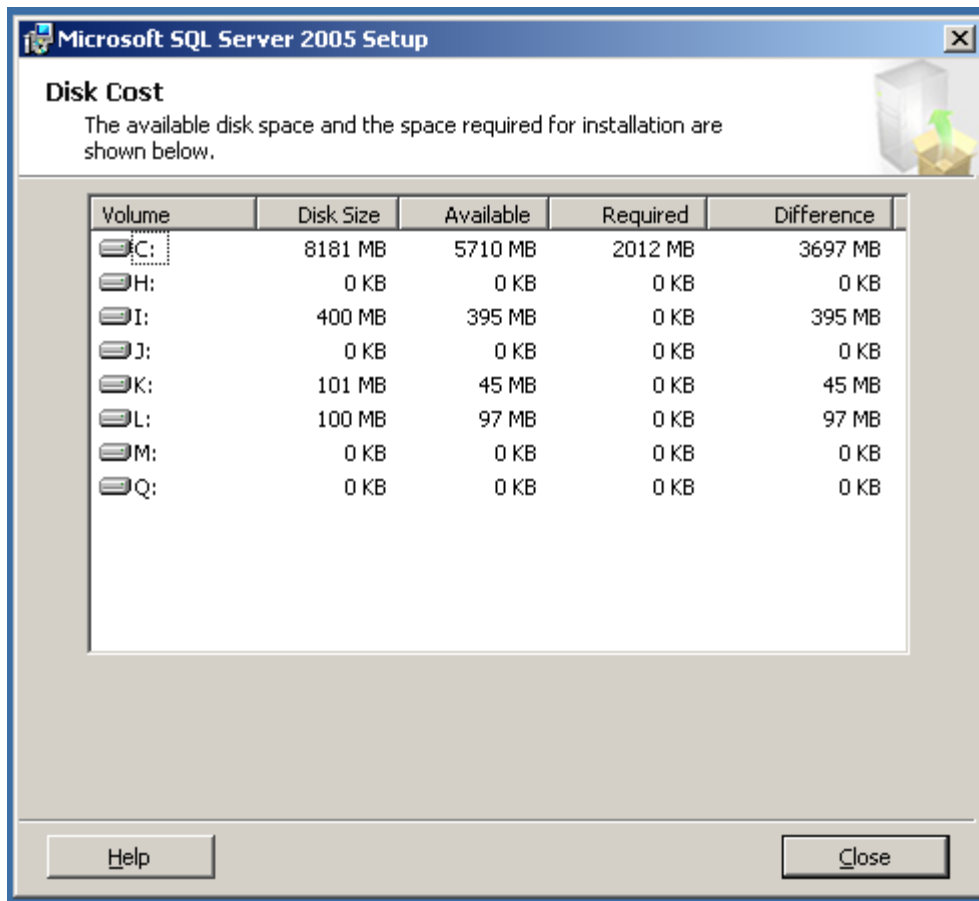
Figure 17. Disk Cost page

10. Select the type of instance that will be installed – either a default or a named instance. If there is already a default instance of SQL Server from an existing installation of SQL Server 2000 in a side-by-side configuration, or another SQL Server 2005 installation, a named instance must be selected. SQL Server 2005 supports a maximum of 25 instances in a server cluster.

   If you are installing a default instance of SQL Server, click **Default instance** and click **Next**.
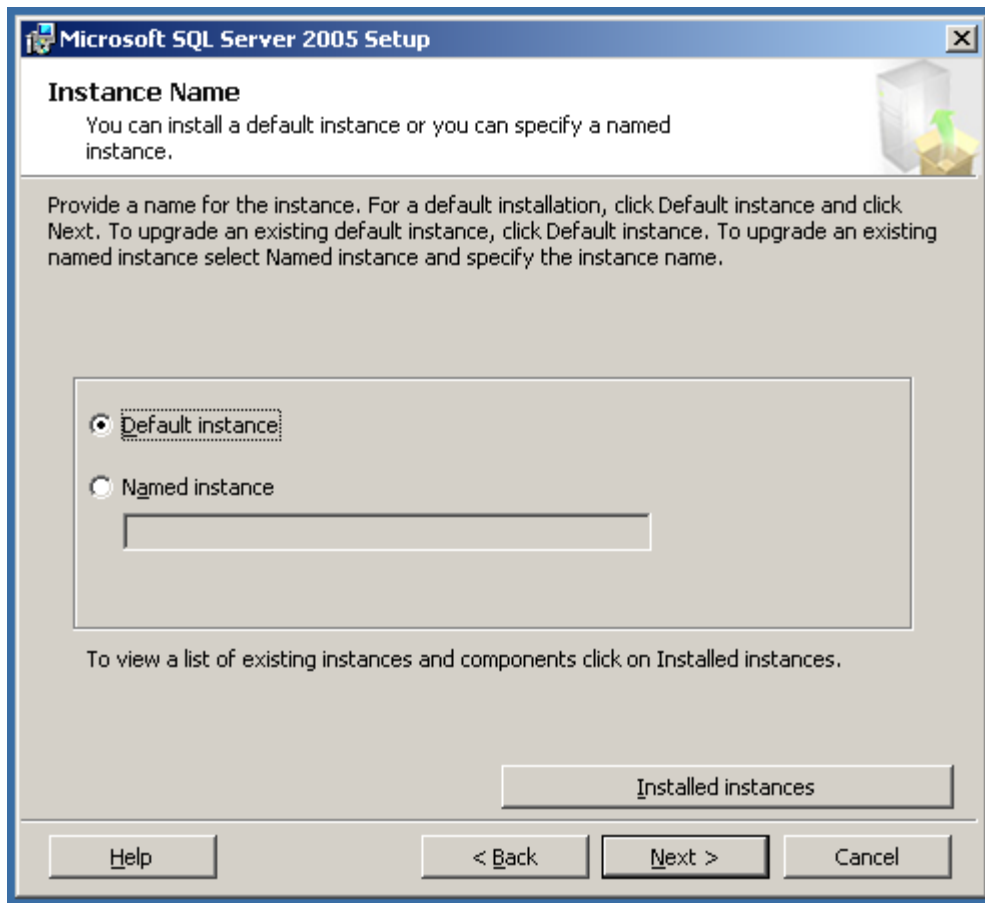
Figure 18. Instance Name page

If you are installing a named instance of SQL Server, click **Named instance**, and enter a name in the box.
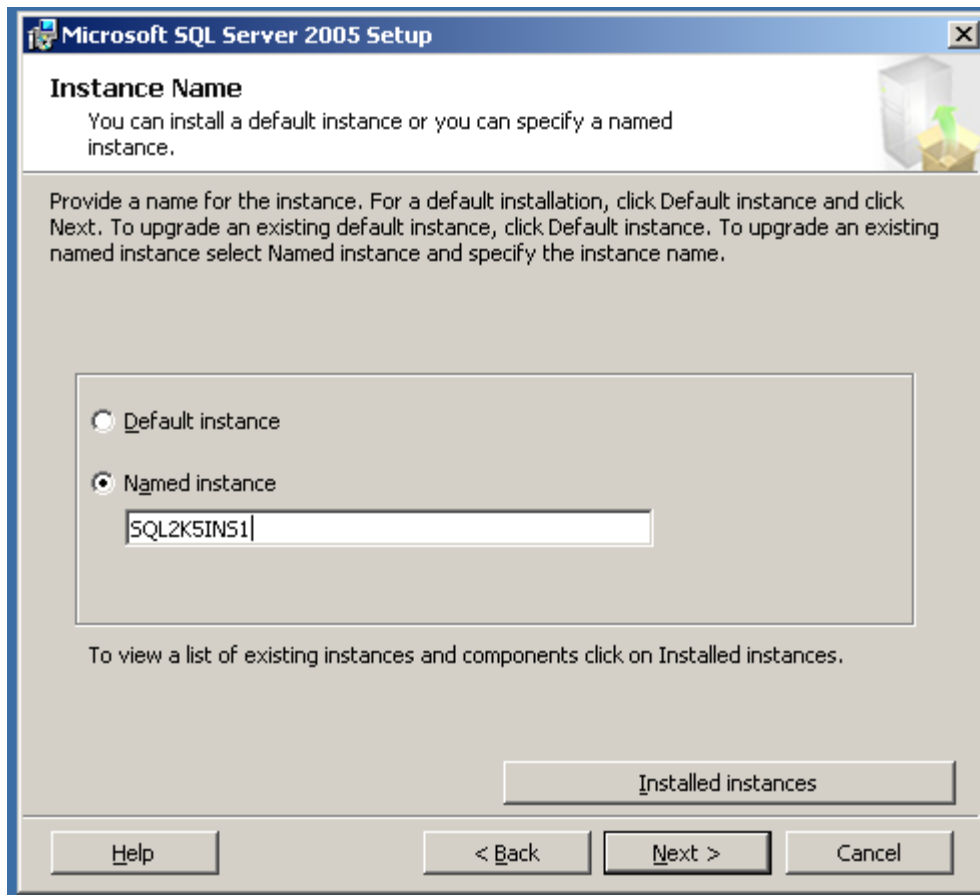
Figure 19. Instance Name page

To see what clustered instances are already installed, click **Installed Instances**.
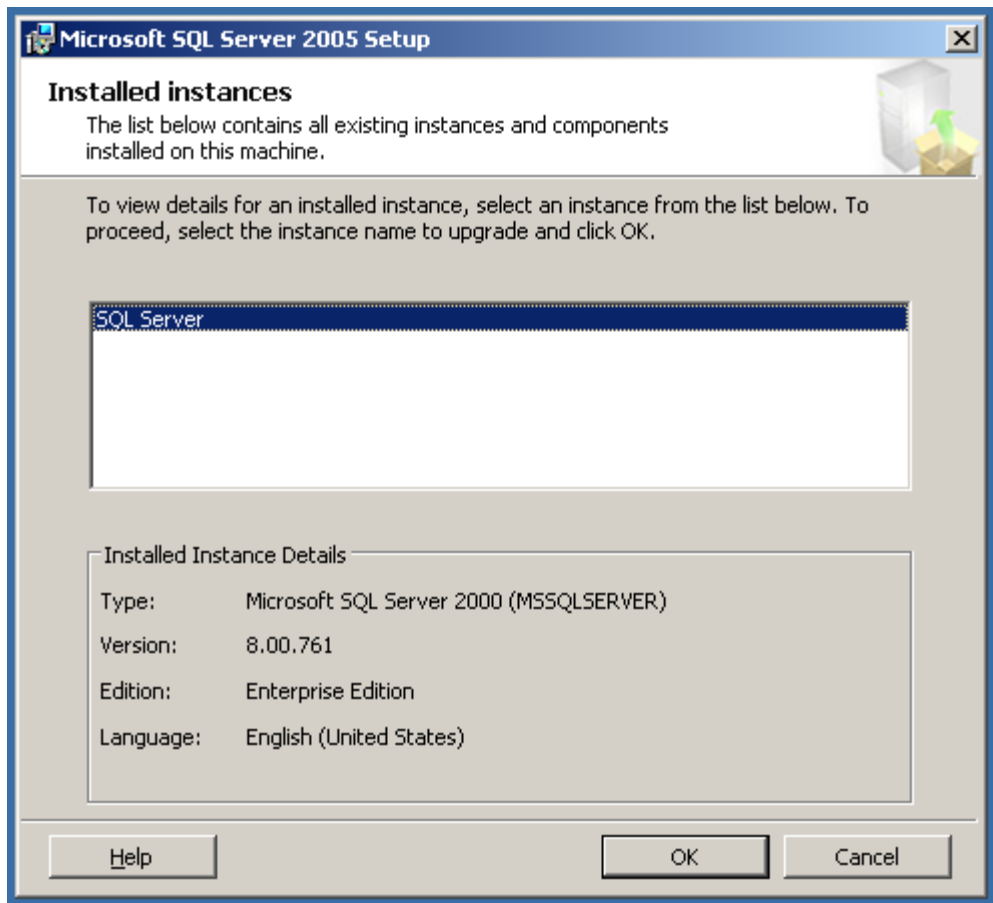
Figure 20. Installed Instances page
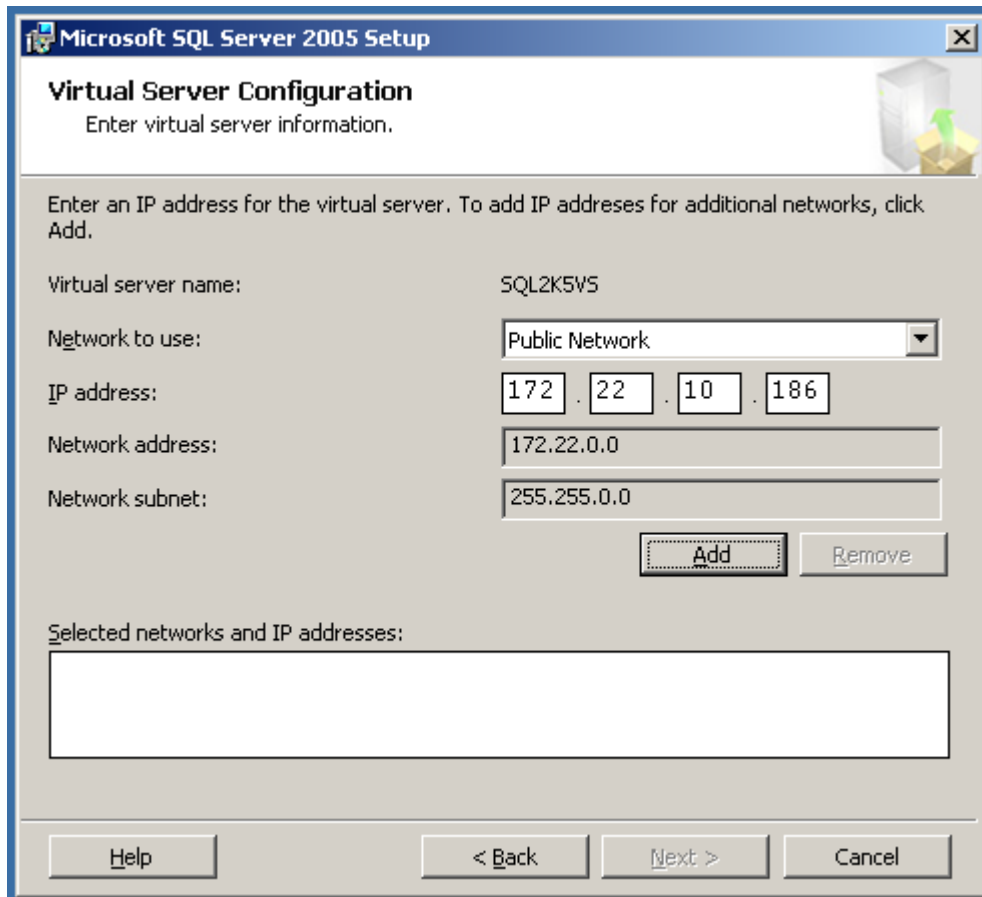
11. On the Virtual Server Name page, enter the name of the SQL Server virtual server. This will be the name used if it is a default instance, or if it is a named instance, the first part before the backward slash. Click **Next** to continue.

> **Note:** Despite the use of the term "virtual server" here, think of this as the failover clustering instance.
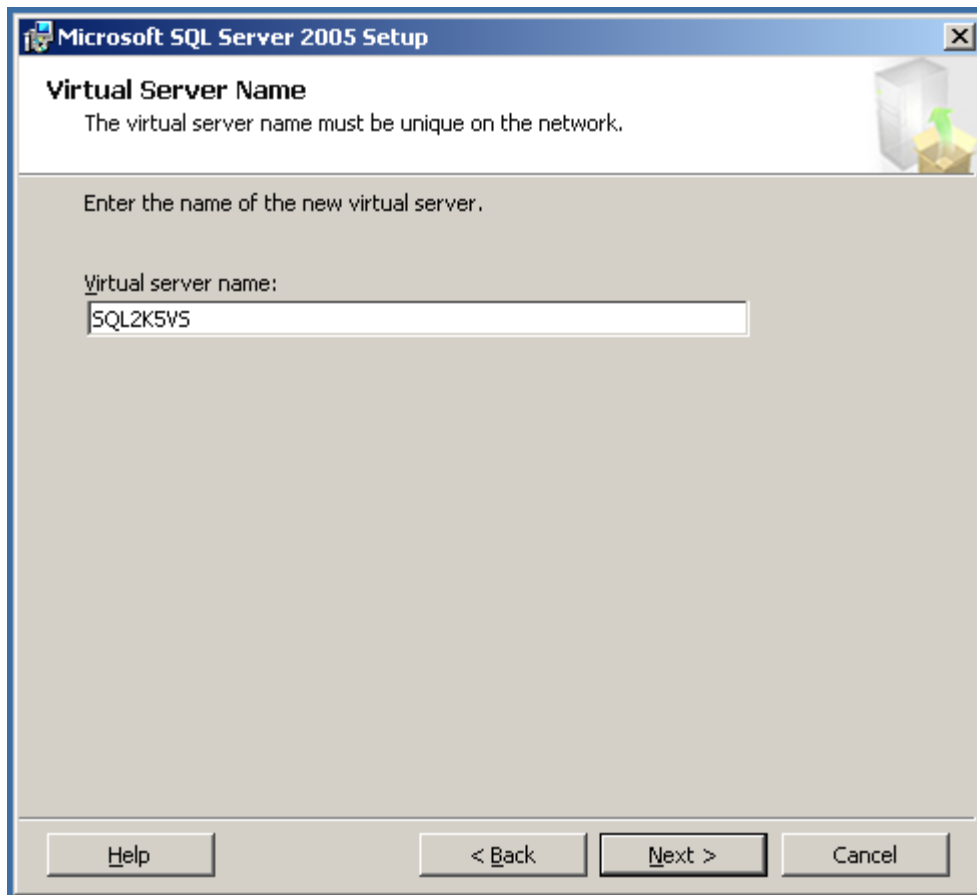
Figure 21. Virtual Server Name page

12. On the Virtual Server Configuration page, first select the network in the **Network to use** list that will be used by SQL Server to allow connectivity from users and applications. This should be an externally (public) facing network in the Windows server cluster. Next, enter the IP address that will be used by SQL Server. Click **Add**.

Figure 22. **Virtual Server Configuration page**

The information will now appear in the **Selected networks and IP addresses** box. SQL Server may be assigned more than one IP address, so if necessary, repeat this step for all IP addresses that SQL Server will use. Click **Next** when finished.

Figure 23. Virtual Server Configuration page

13. On the Cluster Group Selection page, select the cluster group from the **Available cluster groups** list, which contains the disks that this clustered SQL Server installation will use. If an existing failover clustering installation exists on the server cluster already, the group that contains the existing instance will not be available for selection. Do not select the default cluster group, which contains the quorum disk, or the group that contains the Microsoft Distributed Transaction Coordinator.

Once the group is selected, in the **Data files** list, select the disk where the system data files will be placed. Click **Next** to continue.
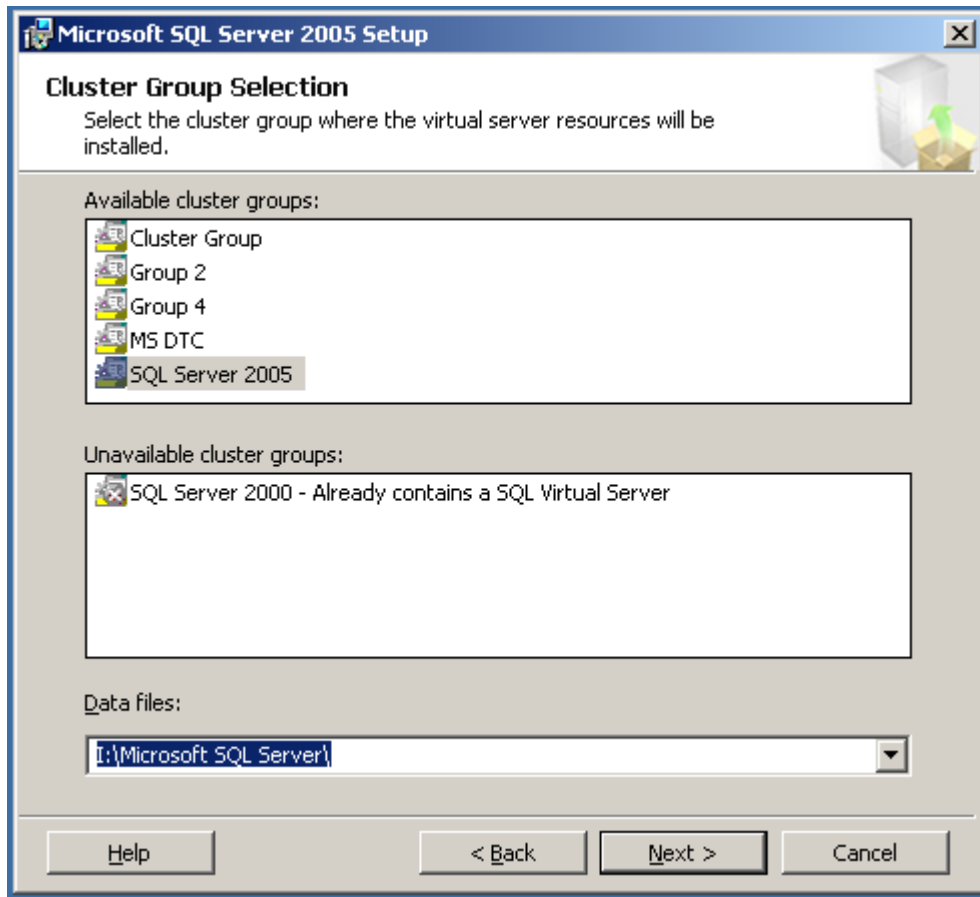
Figure 24. Cluster Group Selection page

14. On the Cluster Node Configuration page, all available nodes will be added to the definition of the failover clustering instance by default. All nodes except the node which you are installing from should be displayed in the **Selected nodes** list. If a node of the cluster is unavailable, it will be displayed in the **Unavailable nodes** list and will have to be added to the failover clustering instance when it is available. If a node should not be part of the definition, select it in the **Selected nodes** list, and click **Remove**. When finished, click **Next**.
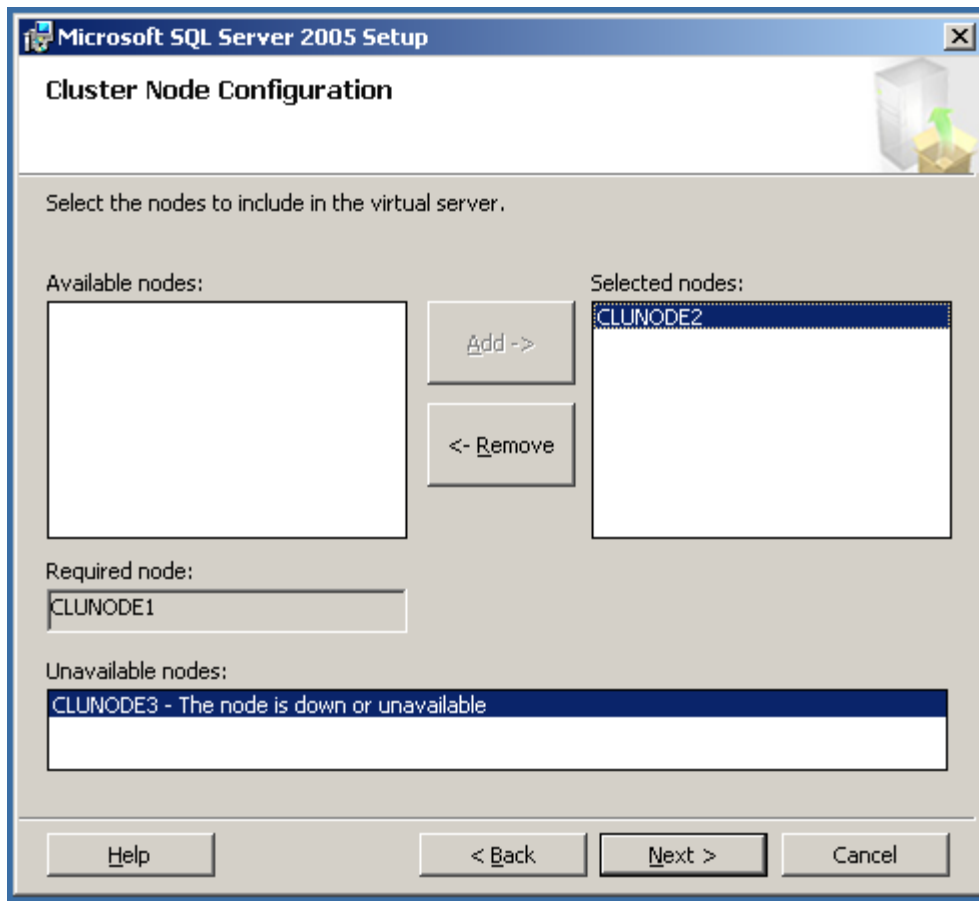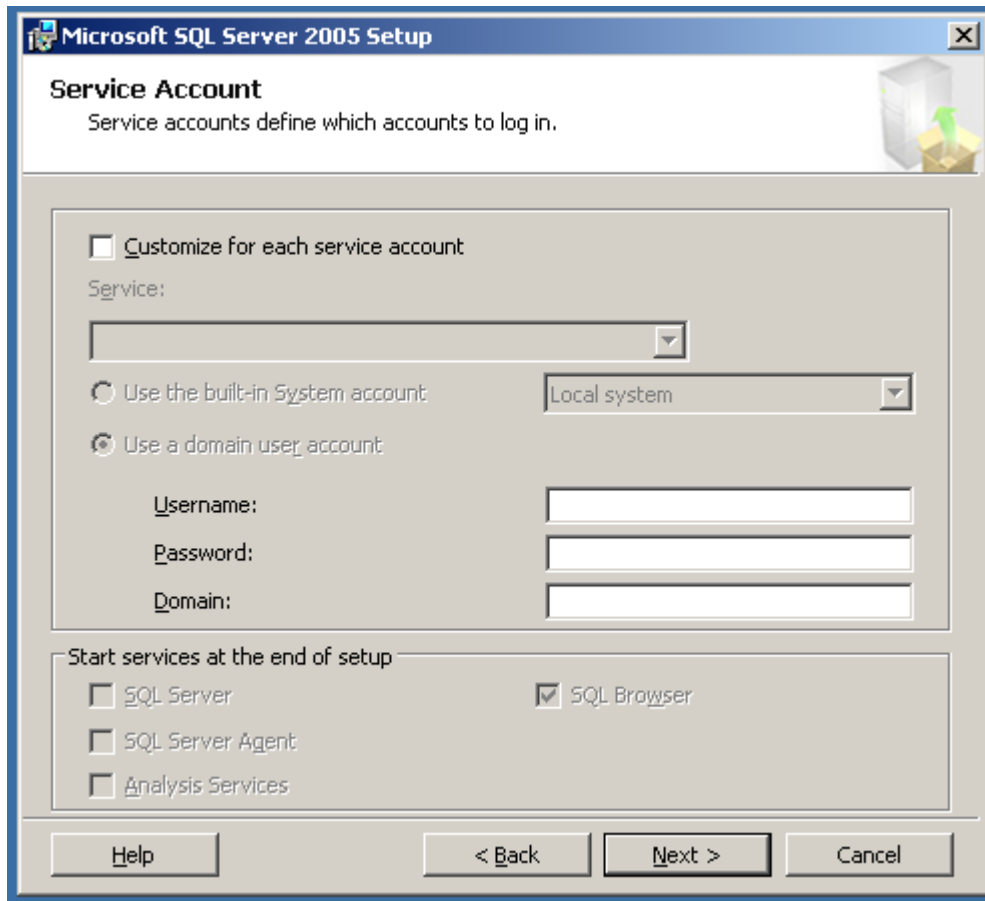
Figure 25. Cluster Node Configuration page

15. On the Remote Account Information page, enter the password of the administrator whom you are logged in as. This administrator must have administrative privileges on all nodes of the cluster. If the user you are currently logged in is not an administrator of the cluster nodes, cancel Setup, log out, and log in as the proper user. This user should not be the SQL Server administrative account. Click **Next**.
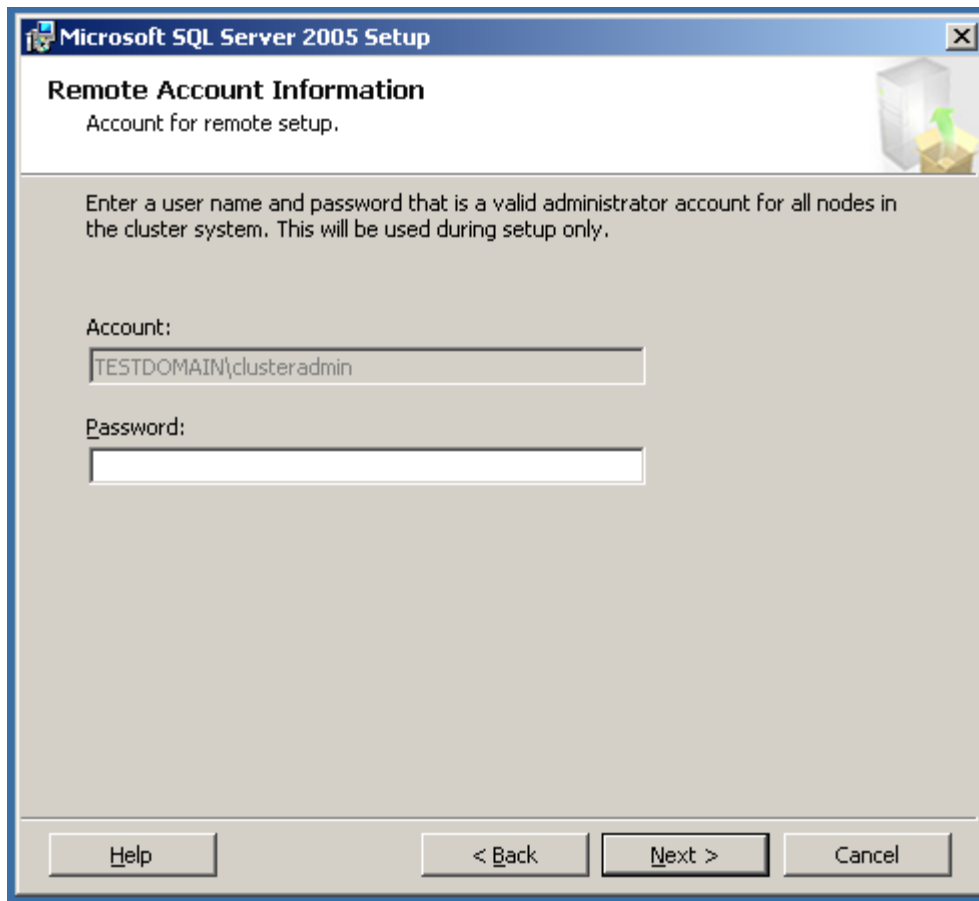
Figure 26. Remote Account Information page

16. The Service Account page configures the service accounts that SQL Server will use after it is installed. You can either use one domain-level account for all services (SQL Server, SQL Server Agent, Analysis Services, and SQL Browser), or you can have a different one for each service. If you wish to use the same account for all, click **Next** to continue.

Figure 27. Service Account page

If you want to customize for each service, select **Customize for each service account**, select the service to modify in the **Service** list, and enter the appropriate information in the **Username**, **Password**, and **Domain** text boxes. Click **Next** when complete.

Figure 28. Service Account page

17. On the Domain Groups for Clustered Services page, select the domain groups that will be used for the failover clustering instance. If you know the name of the groups, enter them in the appropriate text box in the format *DomainName\GroupName*.
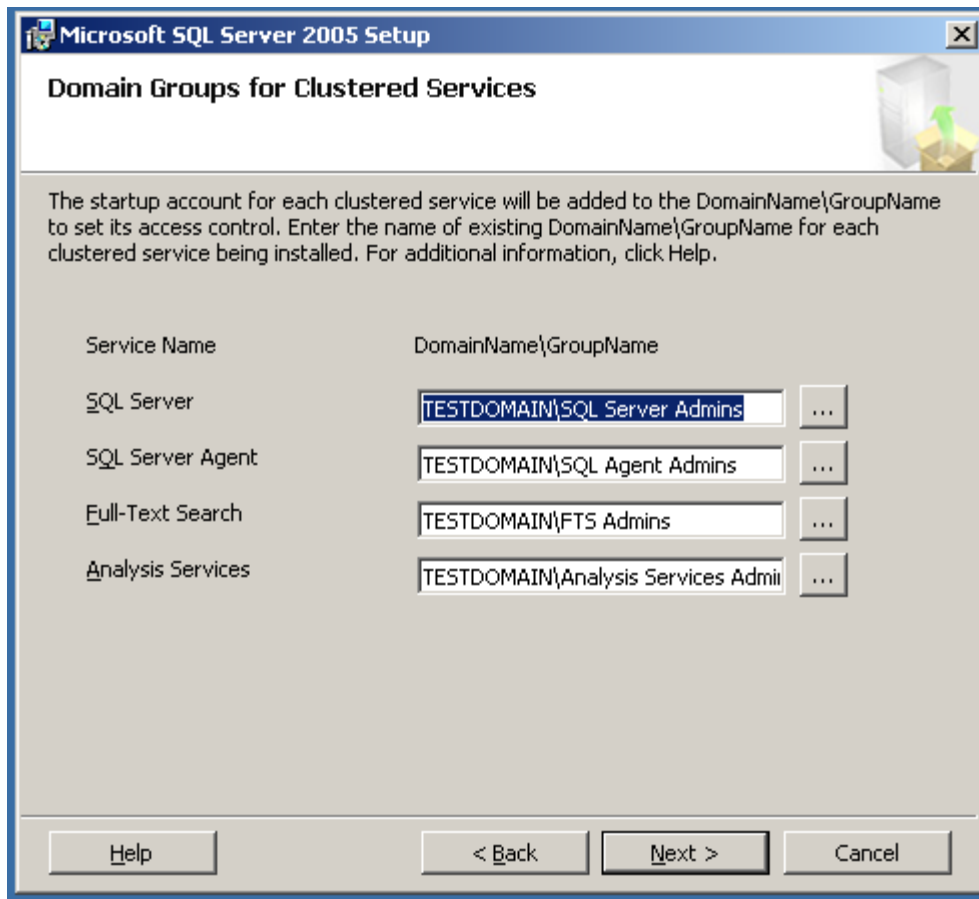
Figure 29. Domain Groups for Clustered Services page

If you do not know the name of the group, click the ellipsis button, and the Select Group page will be displayed. Select the domain to search, enter the name or part of the name, and click **Check Names**. Click **OK**.
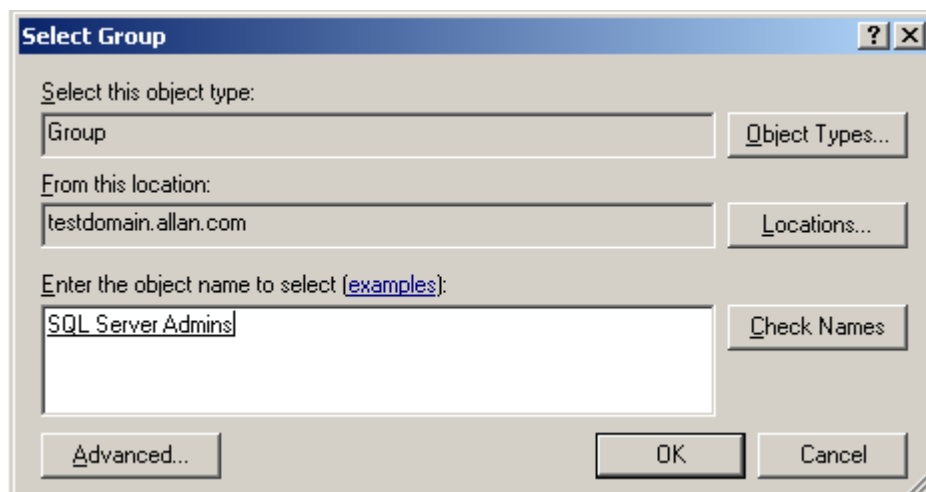
Figure 30. Select Group page

> **Important:** You must enter a domain group for each Service Name listed. The installation process cannot proceed if any entry is left blank. You may use the

same group for all services listed if it has the proper privileges, but it is not recommended.

18. On the Authentication Mode page, click either Windows Authentication Mode or Mixed Mode. Most installations will use Mixed Mode.
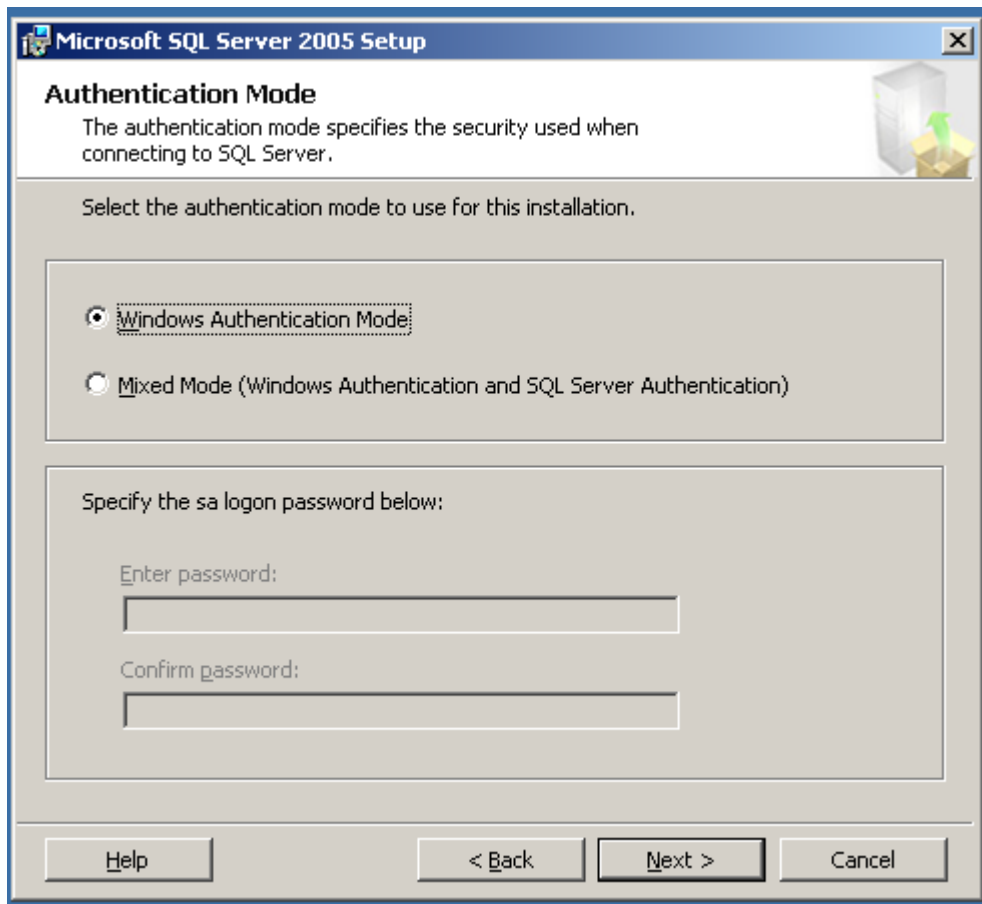
Figure 31. Authentication Mode page

If you select **Mixed Mode**, you will be prompted to enter a password for the **sa** user. You cannot use a blank password for the **sa** user in SQL Server.
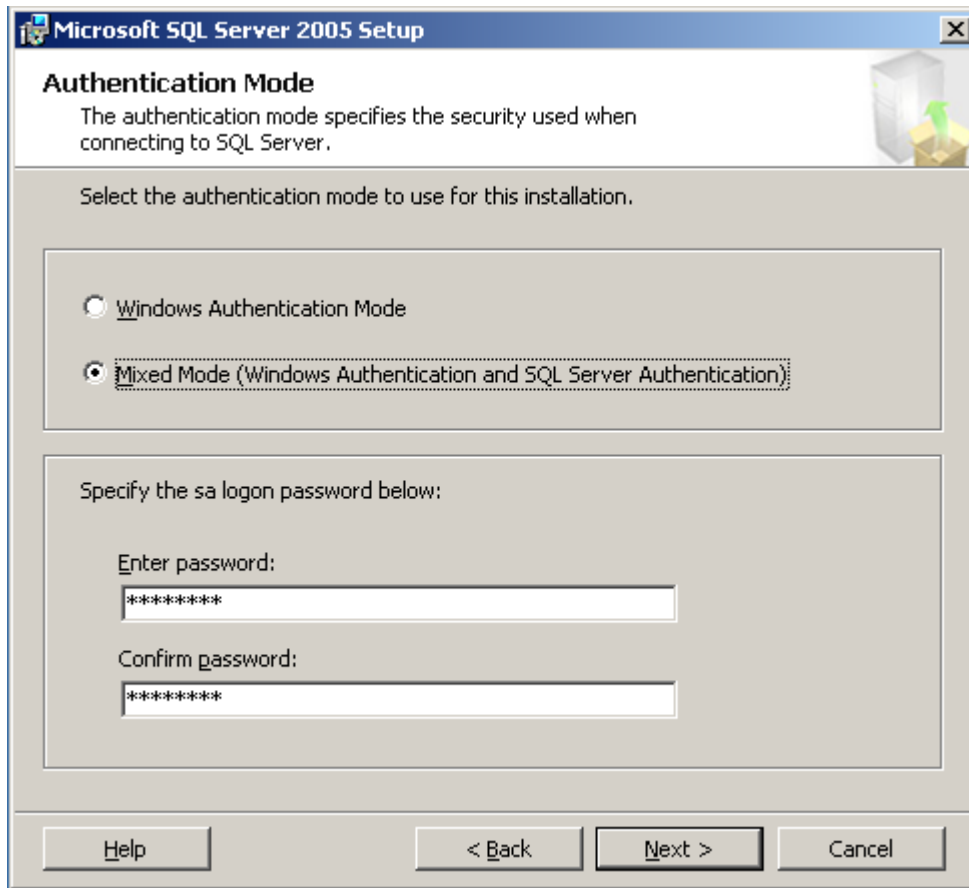
Figure 32. Authentication Mode page

> **Note:** If you select Windows Authentication, the sa user will still exist within SQL Server even though it may not be used. You must set a password for **sa** once the installation process is complete.

19. On the Collation Settings page, if you are installing both SQL Server and Analysis Services at the same time, you use the same collation for both, or use a different one for each. To use the same collation for both, just select the collation either from **Collation designator and sort order**, or for backward compatibility with older databases that may be upgraded, **SQL collations**. To customize for each service account, select the **Customize for each service account** checkbox, and select either **SQL Server** or **SQL Server Analysis Services** in the dropdown list. Click **Next** to continue.
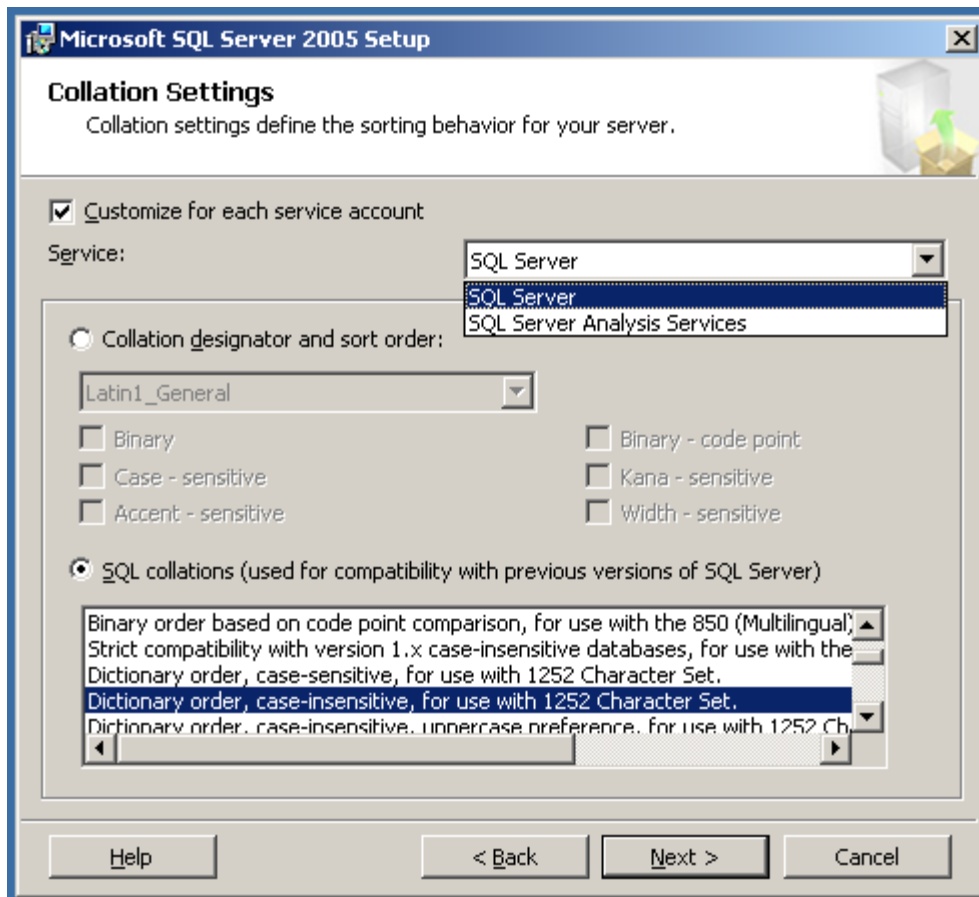
Figure 33. Collation Settings page

20. On the Error and Usage Report Settings page, select the desired options if you wish to send error or usage data back to Microsoft. These are optional. Click **Next**.
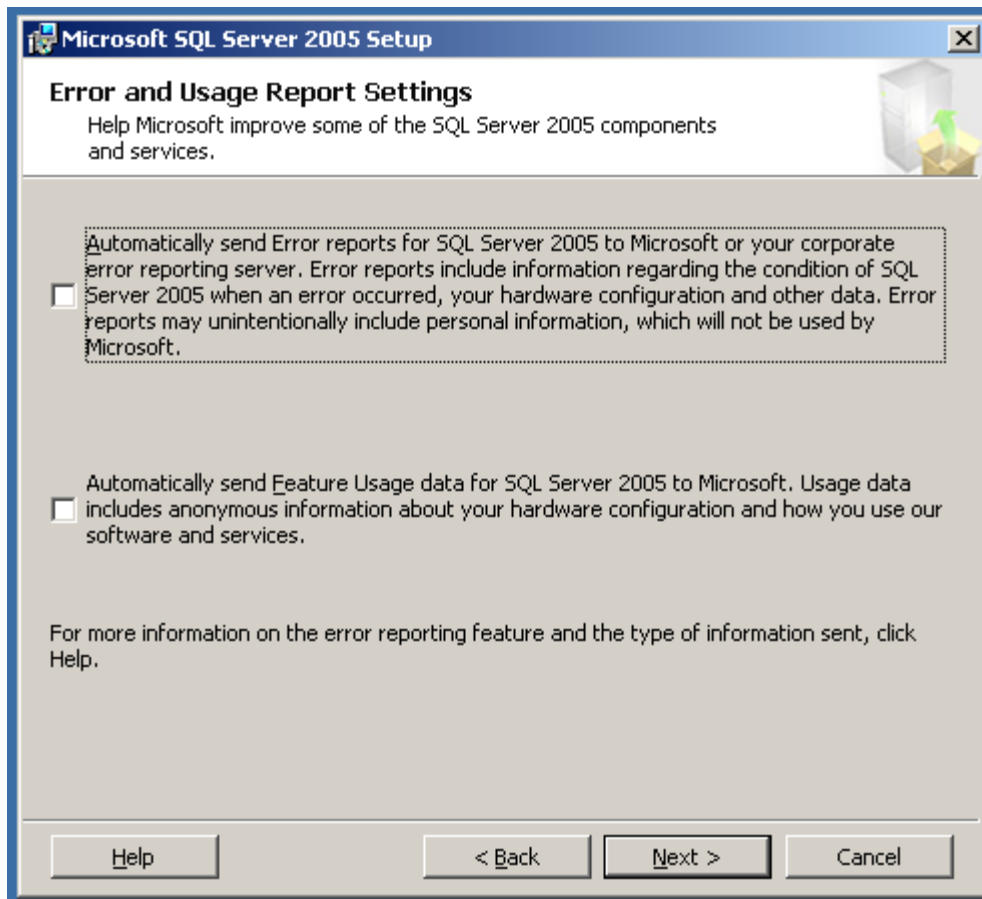
Figure 34. Error and Usage Report Settings page

21. On the Ready to Install page, click **Next** to start the installation process.
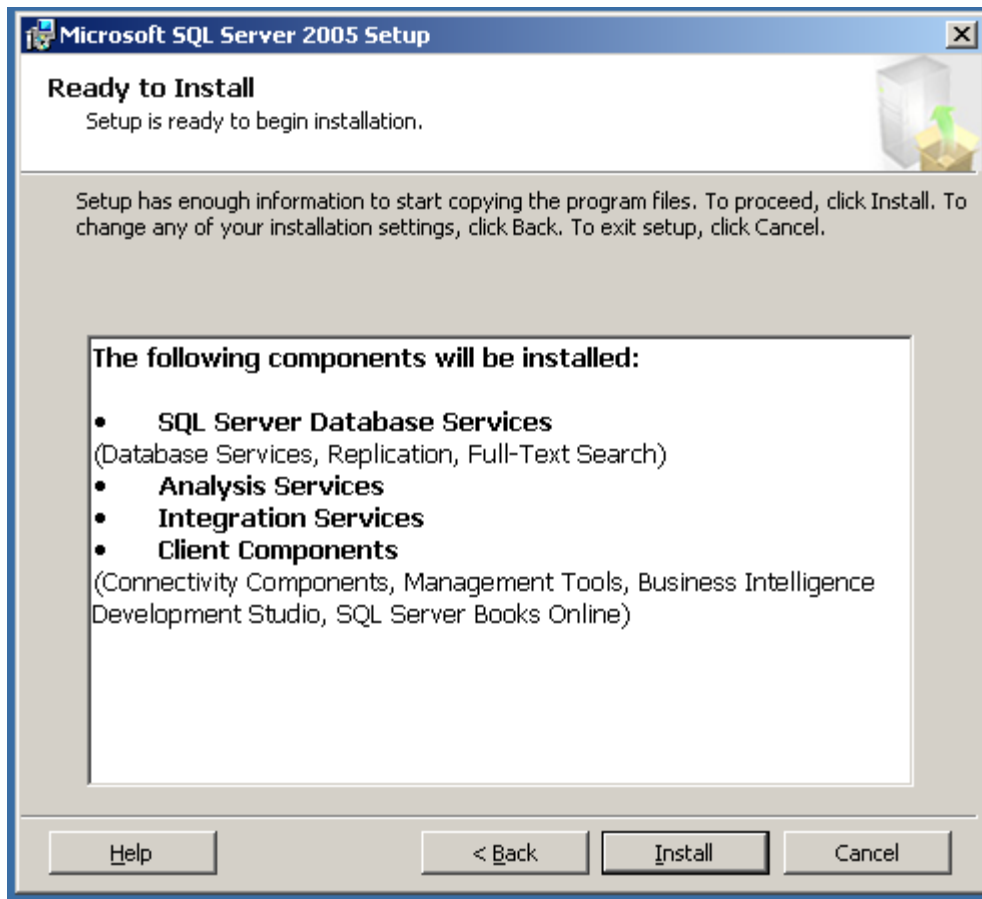
Figure 35. Ready to Install page

Below is the screen you will see as the installation options are configured to install. During this time Setup will be accessing the other nodes to install any prerequisites.
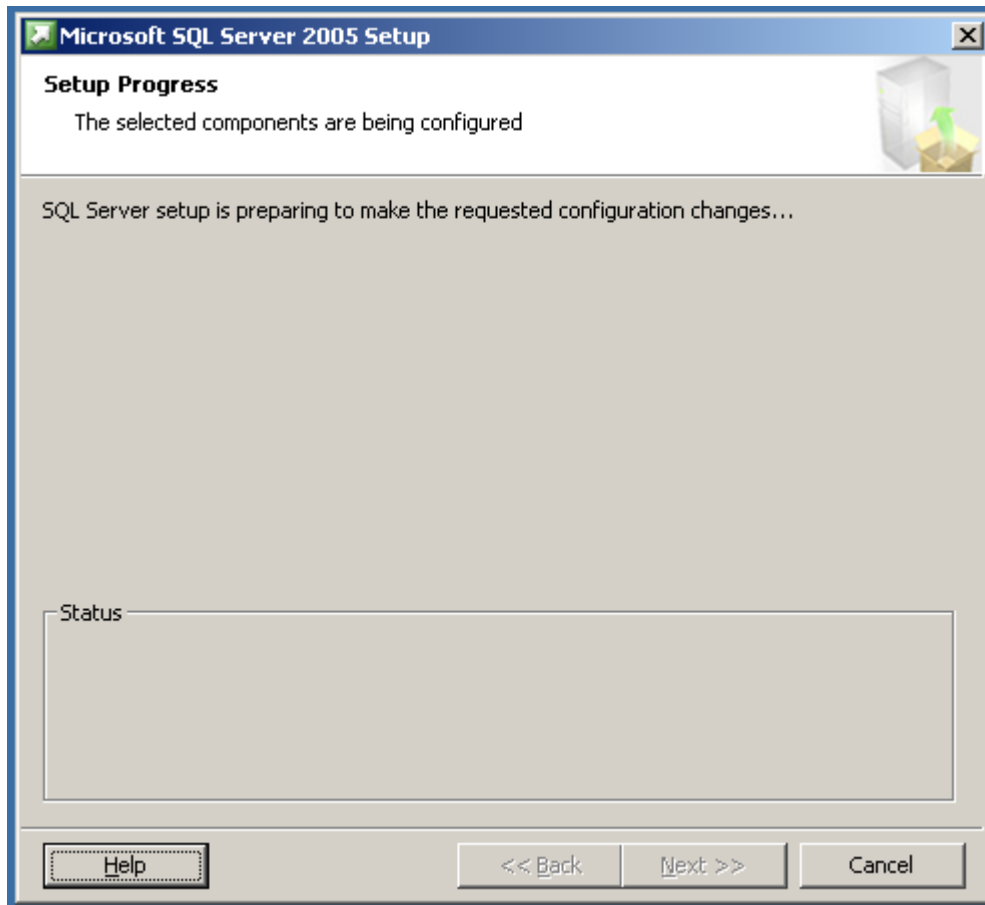
Figure 36. Setup Progress page

22. Once the installation process starts, you can monitor its progress. New to SQL Server 2005 Setup is the ability to see the progress of each node during clustered installs. To see another node, select it from the **Node** list.
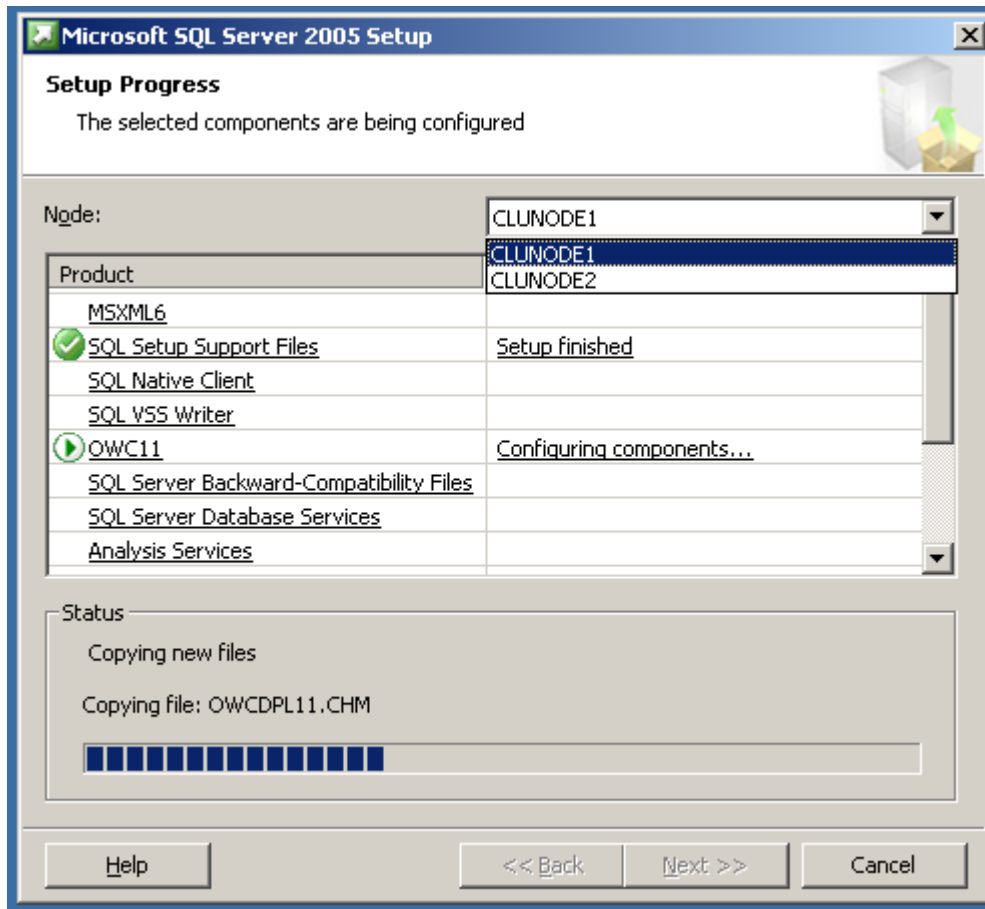
Figure 37. Setup Progress page

23. When setup is complete, the Setup Progress page will show all products with a status of Setup Finished. Click **Next**.
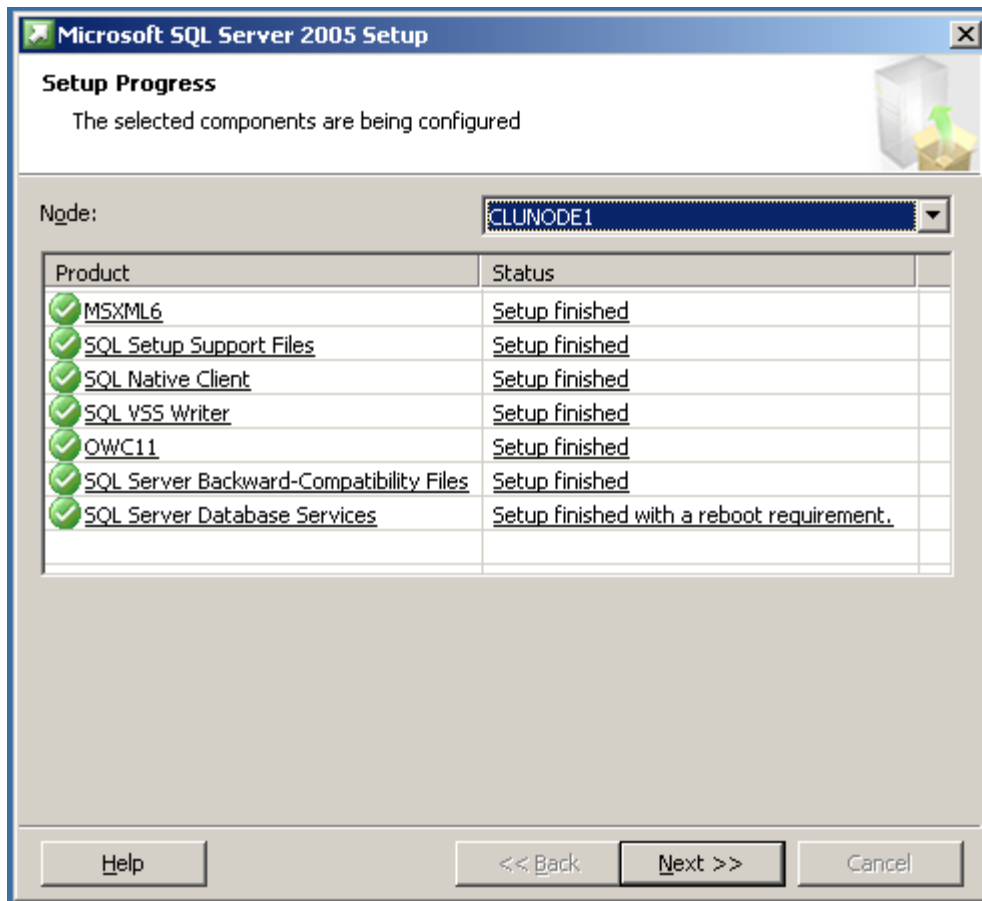
Figure 38. Setup Progress page

24. Read the information presented on the Completing Microsoft SQL Server 2005 Setup page, and click **Finish**.
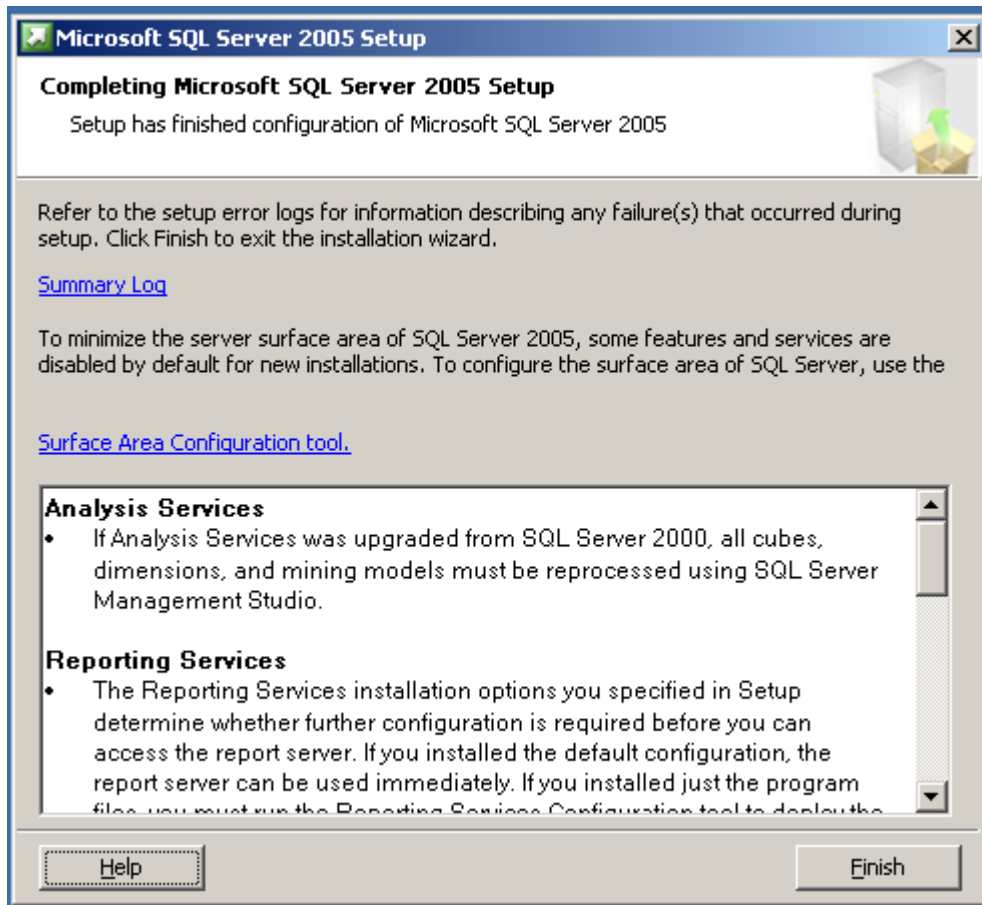
Figure 39. Completing Microsoft SQL Server 2005 Setup page

25. Click **OK** to finish the failover clustering installation process.
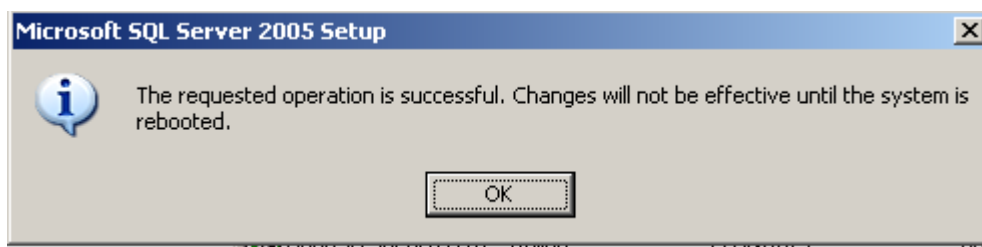
Figure 40.

After rebooting all nodes (not just the node where the installation was launched) if prompted to do so, you can optionally install the SQL Server 2005 client tools on all nodes of the cluster. The steps are similar to those presented here. See How to: Install SQL Server 2005 (Setup) or How to: Install SQL Server 2005 from the Command Prompt.

Appendix B – Upgrading an Existing Failover Clustering Instance to SQL Server 2005 Using Setup

Follow the instructions below to upgrade to a clustered instance of SQL Server 2005 using SQL Server Setup.

26. Log into a node of the Windows server cluster as a domain user with administrative privileges on all nodes of the cluster, such as the user Cluster Administrator.

27. Insert the SQL Server 2005 installation media into the node which you logged into, or navigate to the proper network location at your business. If you have autorun enabled on your CD or DVD drive, the following menu below should appear. If the screen does not appear, navigate to the Servers directory of the media and run setup.exe. Click **Server components, tools, Books Online, and samples**.

Figure 41. Microsoft SQL Server 2005 Setup Start page

28. Read the End User License Agreement (EULA). Select **I accept the licensing terms and conditions** and click **Next**.
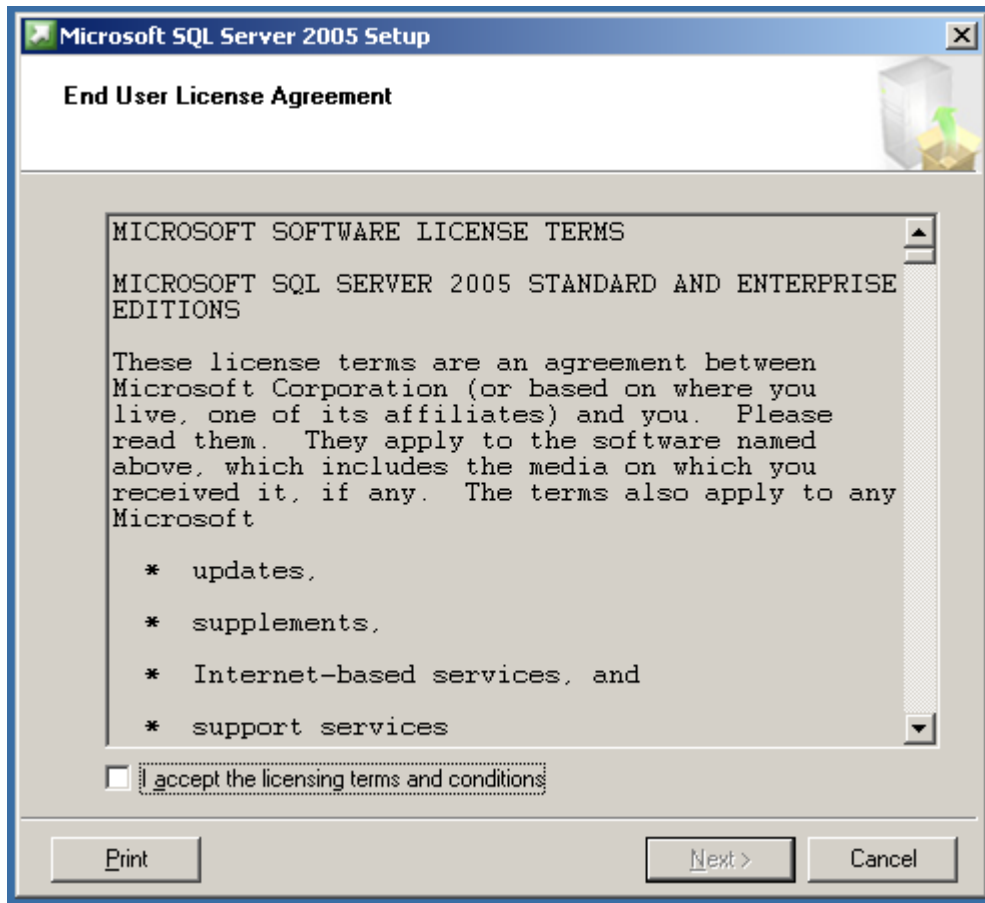
Figure 42. End User License Agreement

29. Setup will now install any prerequisites on the node which started the install process. During the process, a green checkmark will appear next to completed tasks and a red arrow pointing right will indicate the current task being executed. When all tasks are completed, click **Next** to continue.

> **TIP:** To speed up the initial phase of the installation or upgrade process, we recommend installing the proper version of the .NET Framework (which can be downloaded from the Microsoft .NET Framework Developer Center) on each node before starting the SQL Server installation so that SQL Server Setup will not have to do it.
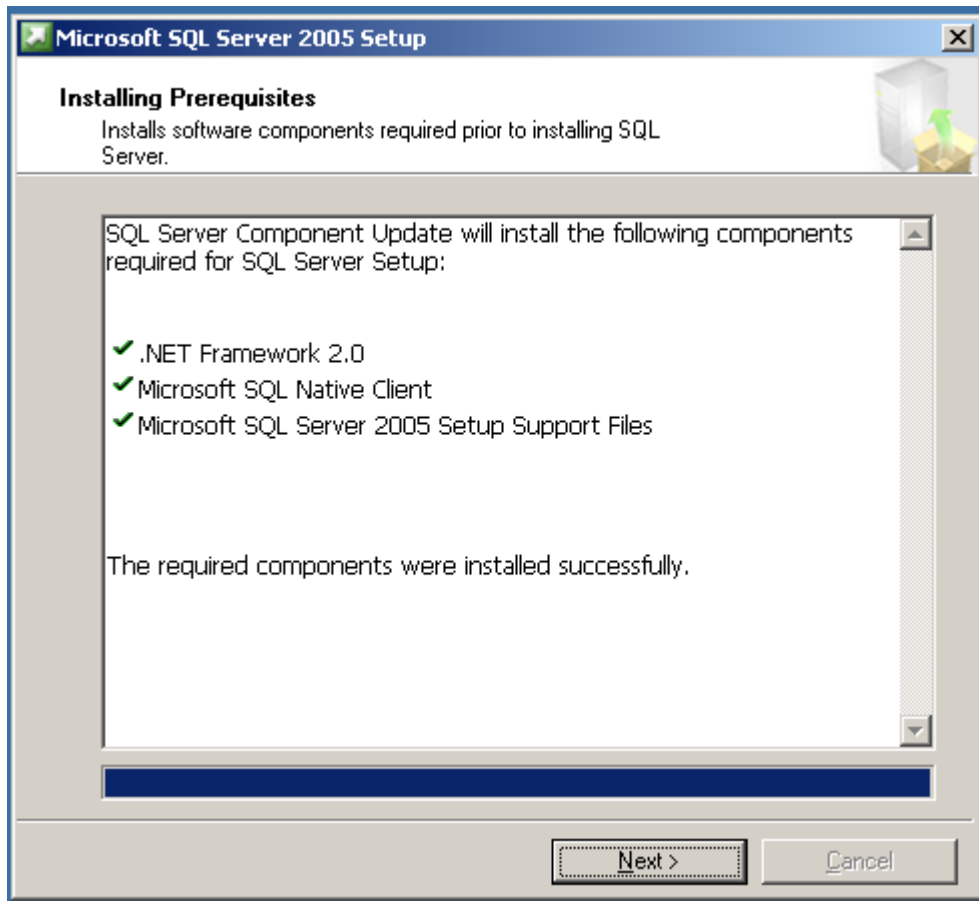
Figure 43. Installing Prerequisites page

Setup will now perform a configuration check of the node that started the installation process.
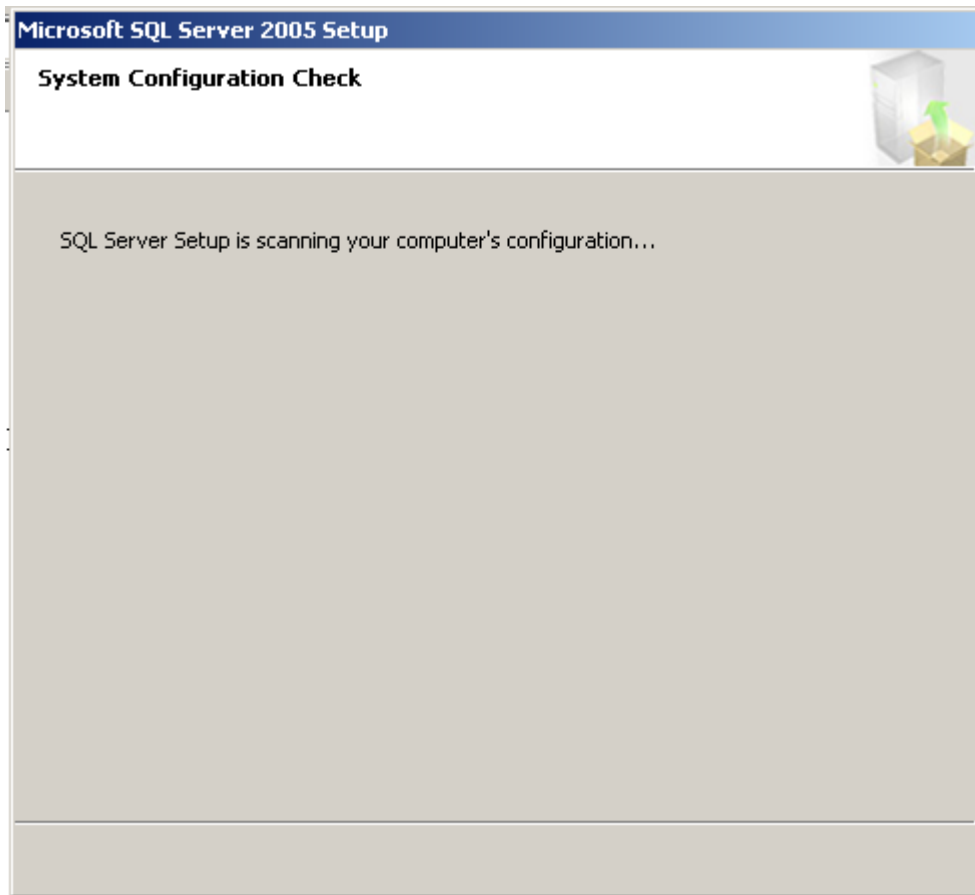
Figure 44. System Configuration Check page

30. When the Welcome to the Microsoft SQL Server Installation Wizard page appears, click **Next**.

Figure 45. Welcome to the Microsoft SQL Server Installation Wizard page

31. Setup will now do a more exhaustive check of the node which started the installation process. Any warnings or errors will be detected, and they can be seen by clicking **Messages** next to the appropriate item, or by viewing a report via the **Report** dropdown. The report can be viewed on the screen, saved to a file, copied to the Clipboard, or e-mailed.

> **Note:** If you have any errors, fix them before proceeding with the installation as they may either cause the installation to fail or compromise the stability of the installation. Exit out of Setup and rerun when complete. Warnings are not fatal, but they should be evaluated to see if they are valid and need to be fixed.
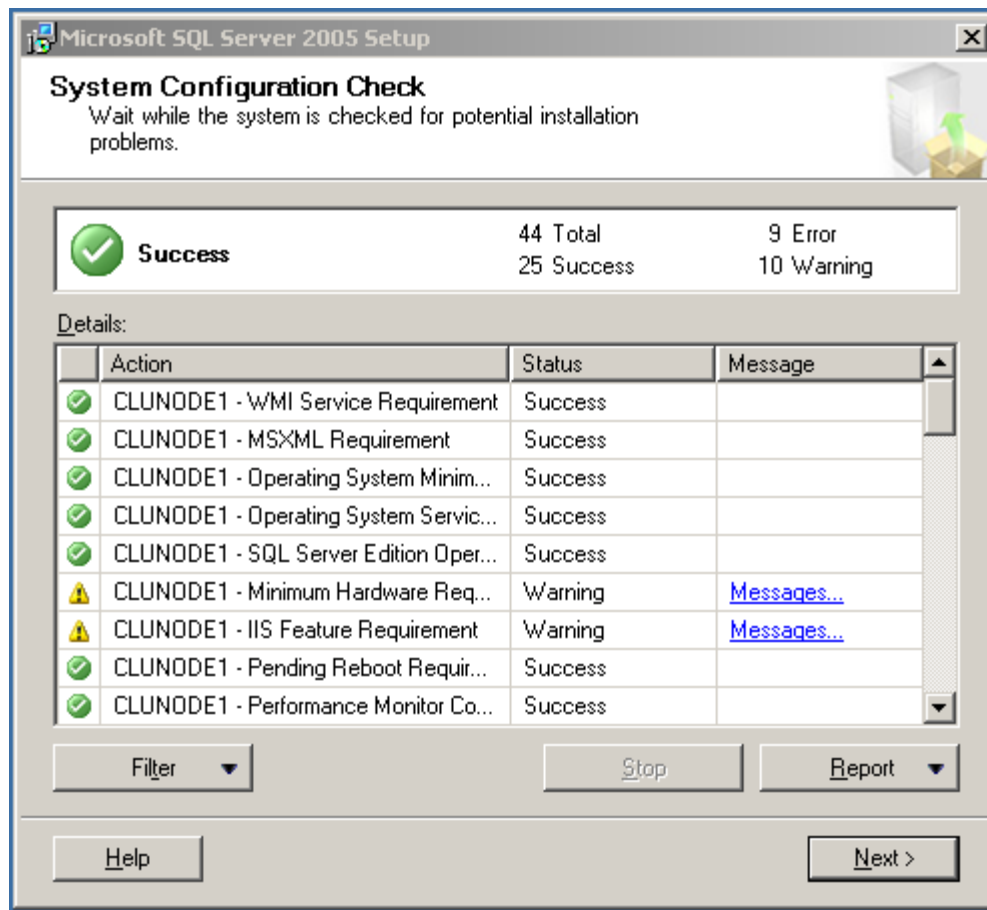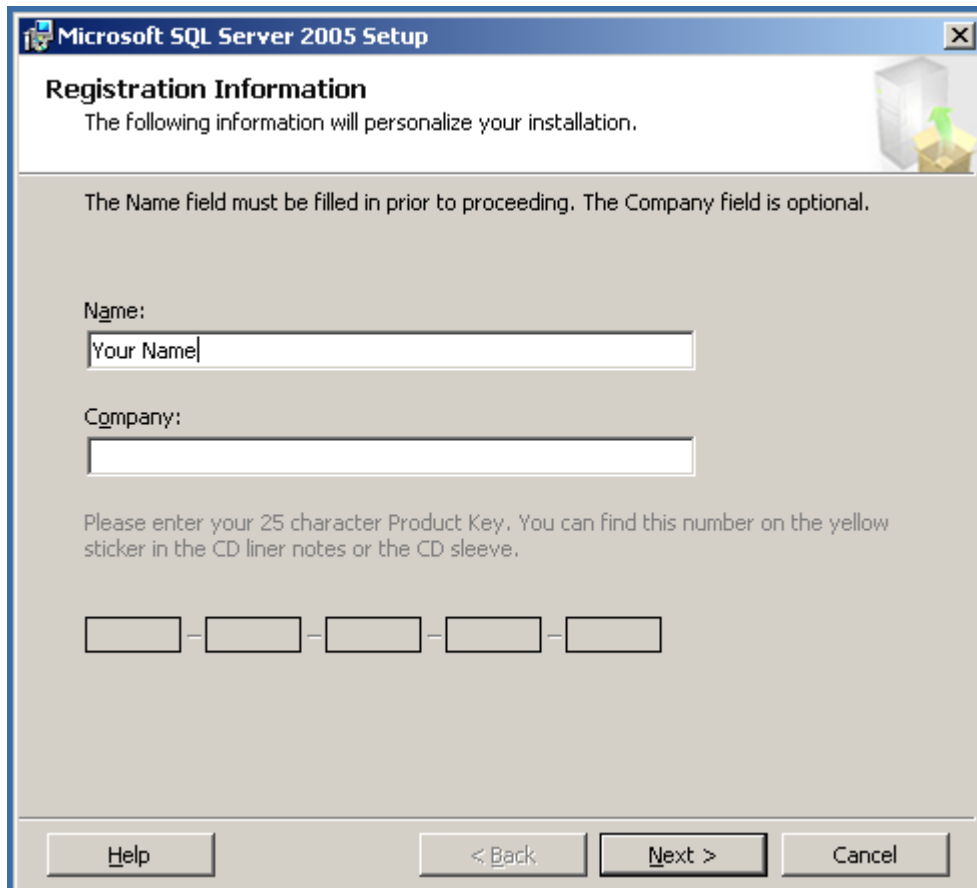
Figure 46. System Configuration Check page

32. When the check is complete, click **Next**. Setup will prepare the installation.
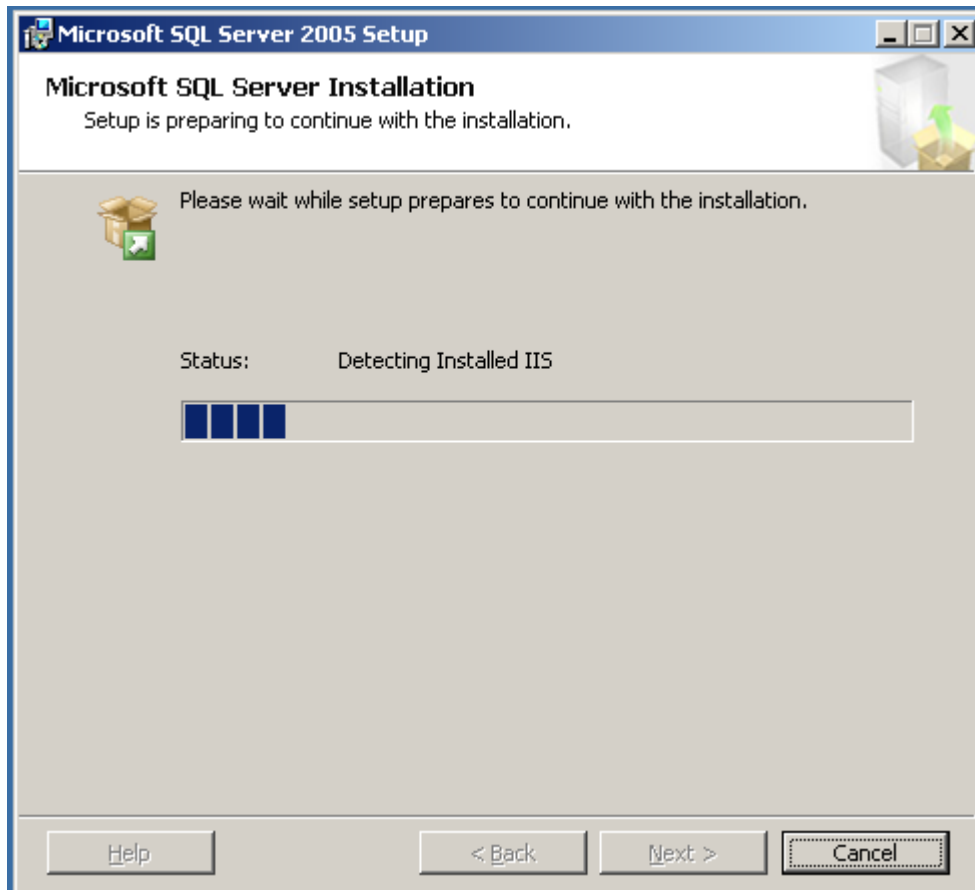
Figure 47. Microsoft SQL Server Installation page

33. Enter your name and a valid Product Key (PKD) from your installation media. Click **Next**.

**Figure 48. Registration Information page**

34. Select the components that you want to install during the Setup process. You do not need to select all features, as they can be added later. To install a relational engine failover clustering instance, select the **SQL Server Database Services** check box and then select the **Create a SQL Server failover cluster** check box. If you do not select the latter, it will be installed as a stand-alone instance. To install an Analysis Services failover clustering instance, select the **Analysis Services** check box and then select the **Create an Analysis Services failover cluster** check box. If you do not select the latter, it will be installed as a stand-alone instance. Click **Next** when done.

Figure 49. Components to Install page

If you click **Advanced**, the Feature Selection page will be displayed. Here you can further customize your SQL Server installation; however it is recommended that you not change any of the options unless you are familiar with the workings of SQL Server 2005. These screens are described here so you can see what is done on them should you choose the Advanced option.
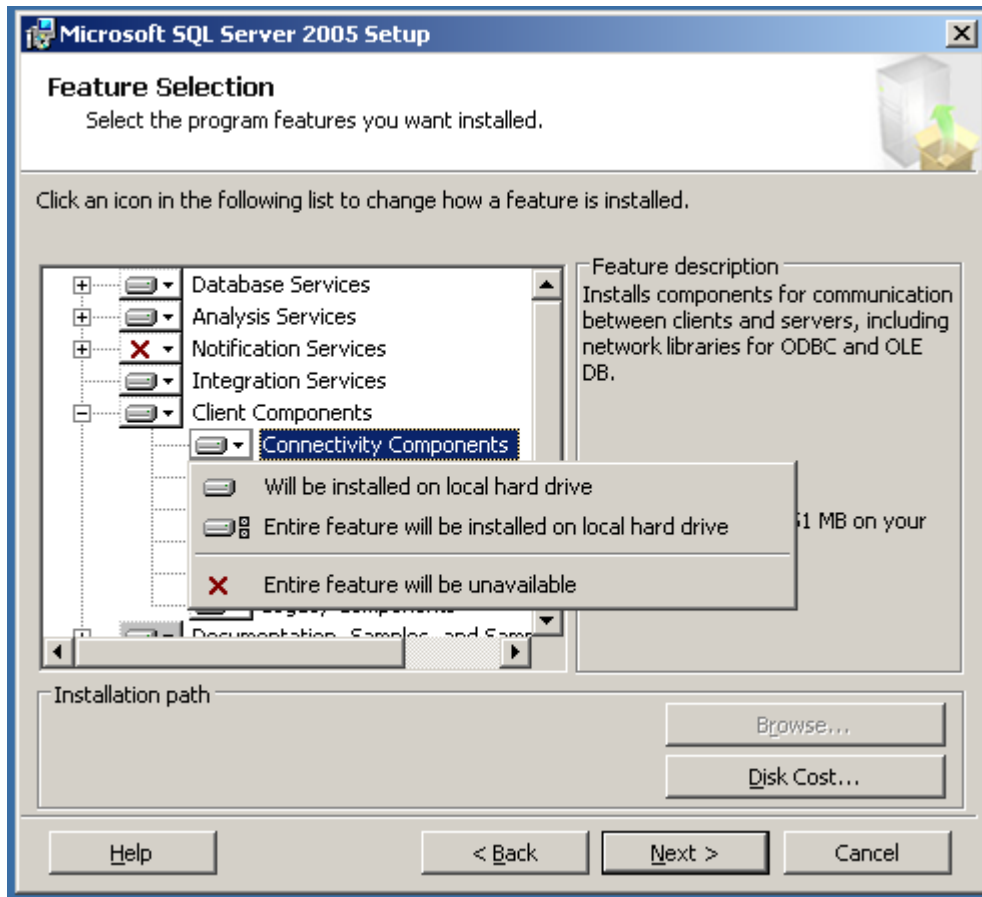
Figure 50. Feature Selection page

If you click **Browse**, the Change Folders page appears. On this screen you can change the top-level installation folder where the SQL Server program files will be installed on the nodes.
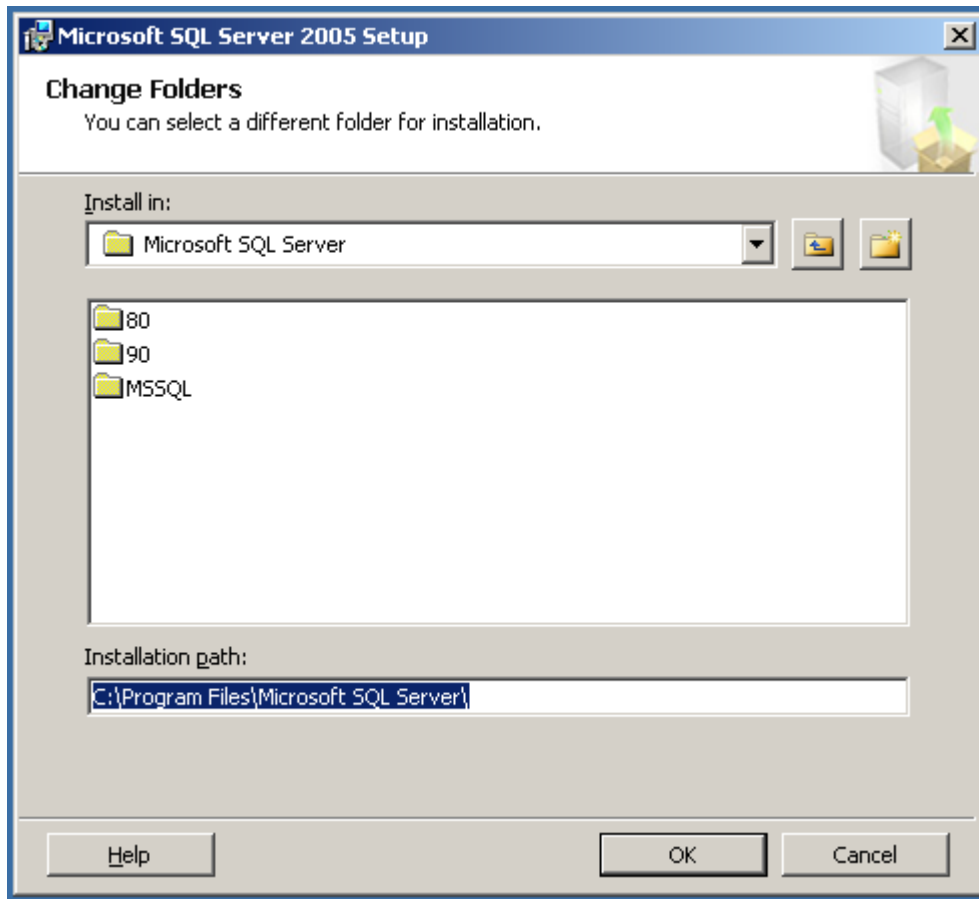
Figure 51. Change Folders page

If you click **Disk Cost**, Setup will display the disks available, how much space is left, and how much space is needed by SQL Server for the installation to complete successfully.
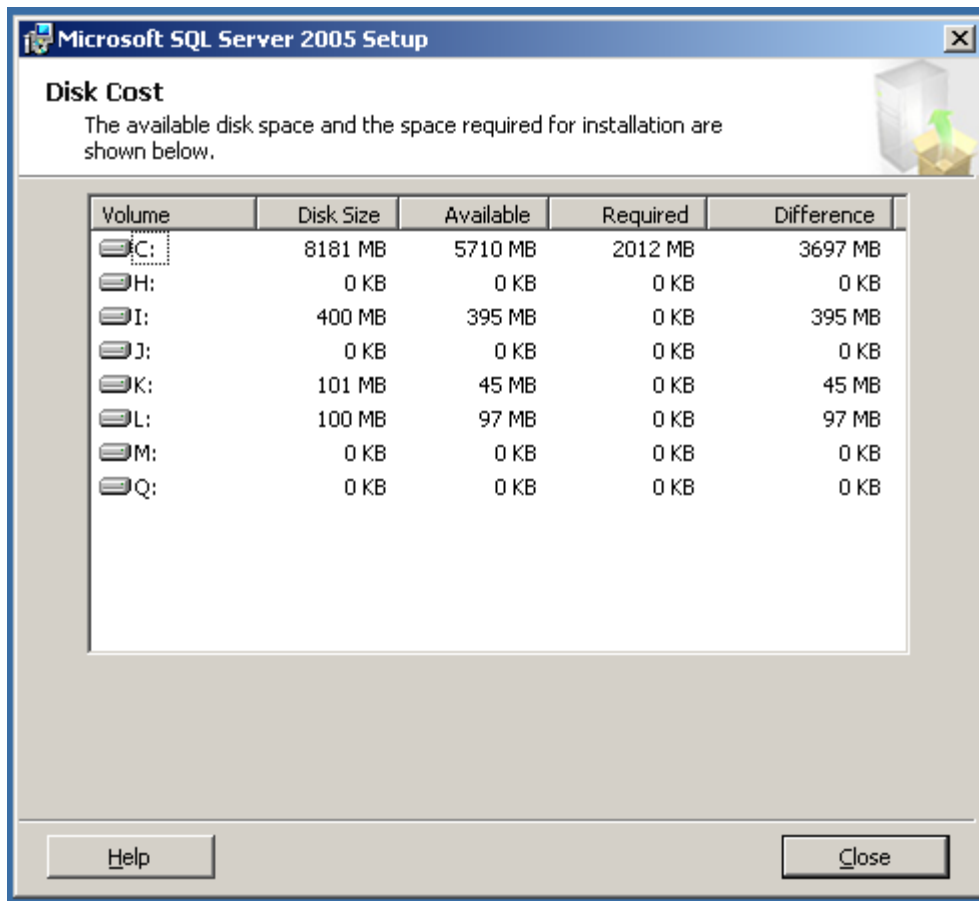
Figure 52. Disk Cost page

35. If you are upgrading a default instance of SQL Server, click **Default instance.** If you are upgrading a named instance of SQL Server, click **Named Instance**, and then enter the existing instance's name in the text box. If you do not know what clustered instances are already installed, click **Installed Instances**. The Installed Instances page will appear. Click **Next** when finished. .
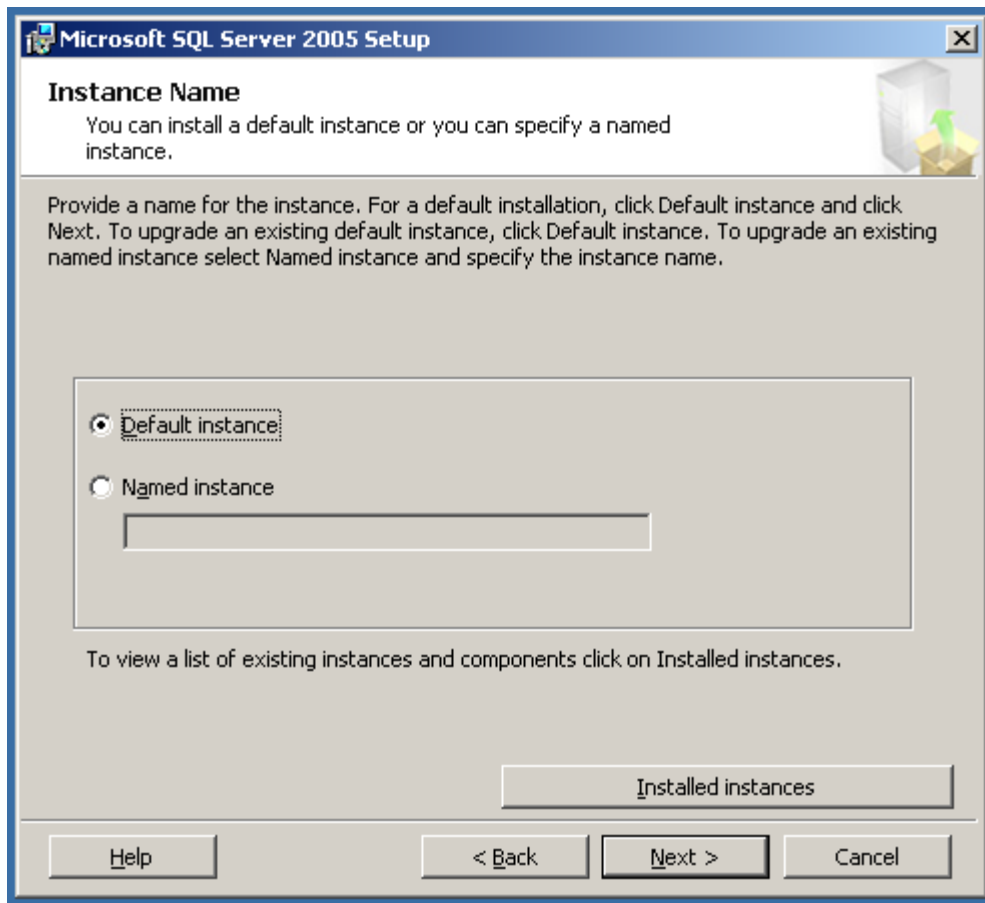
Figure 53. Installed Instances page

36. On the Existing Components page, select the check box next to the instance that you selected to upgrade on the previous screen.
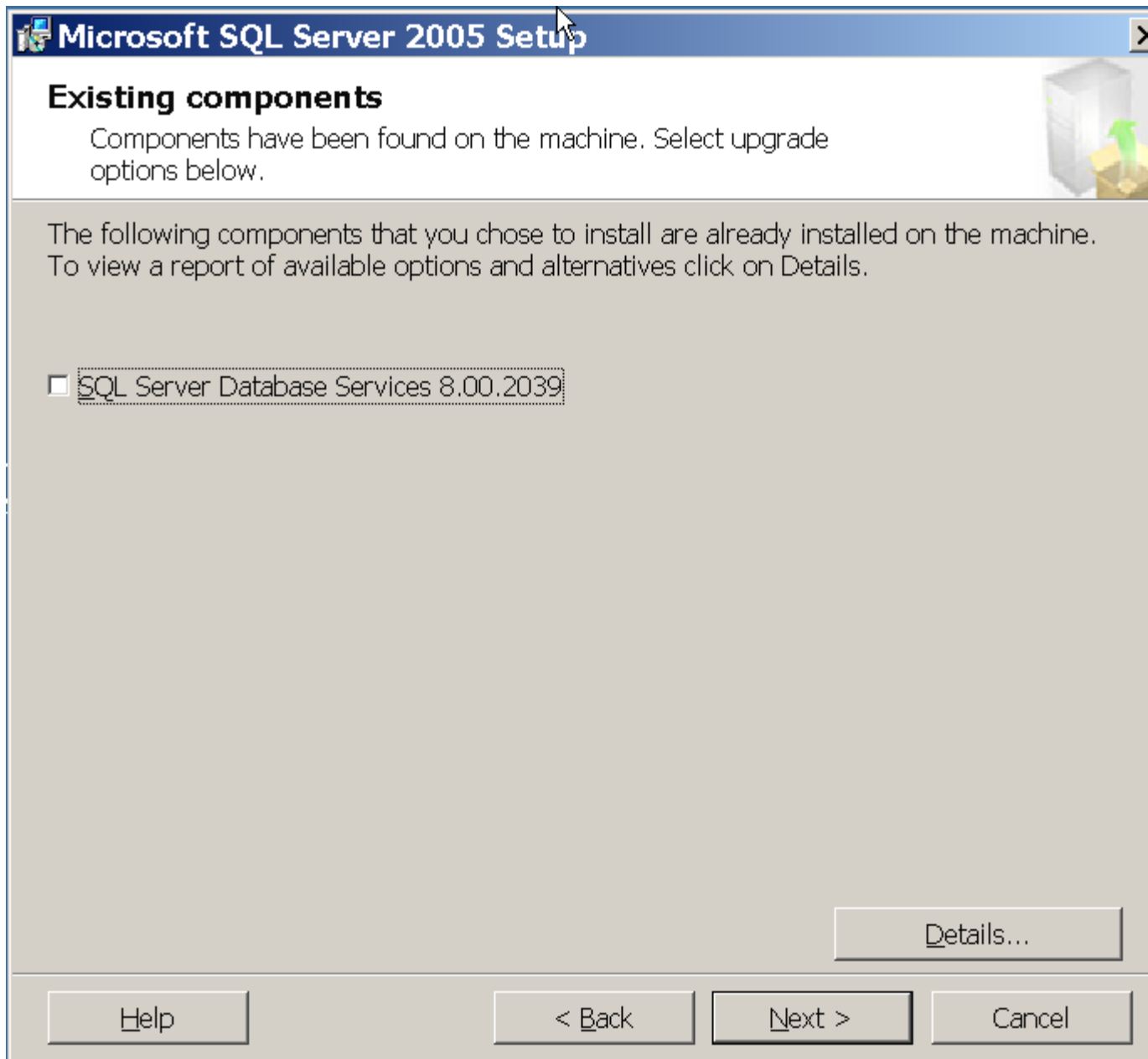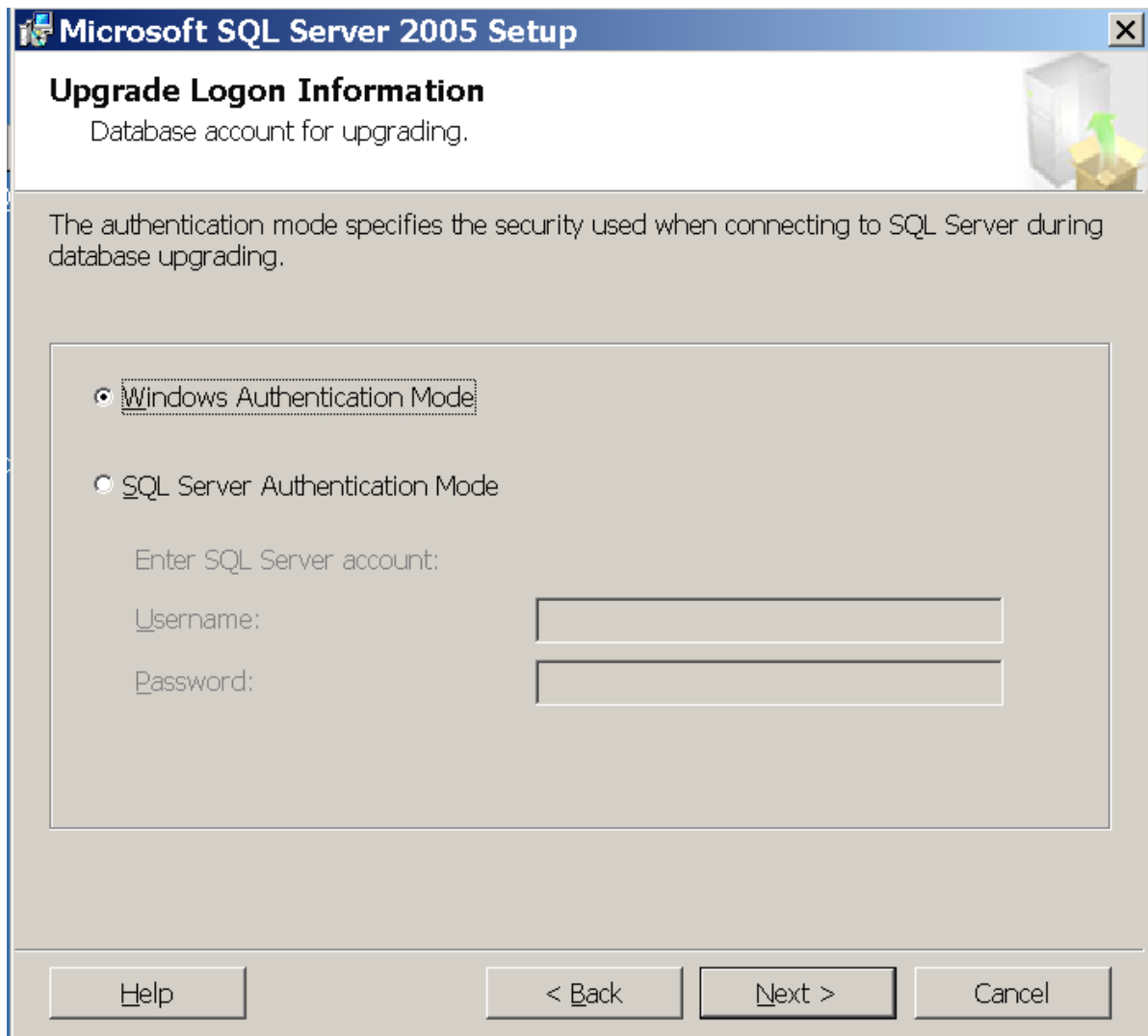
Figure 54. Existing Components page

To see information about the upgrade, click **Details**. Otherwise, click **Next** to continue.
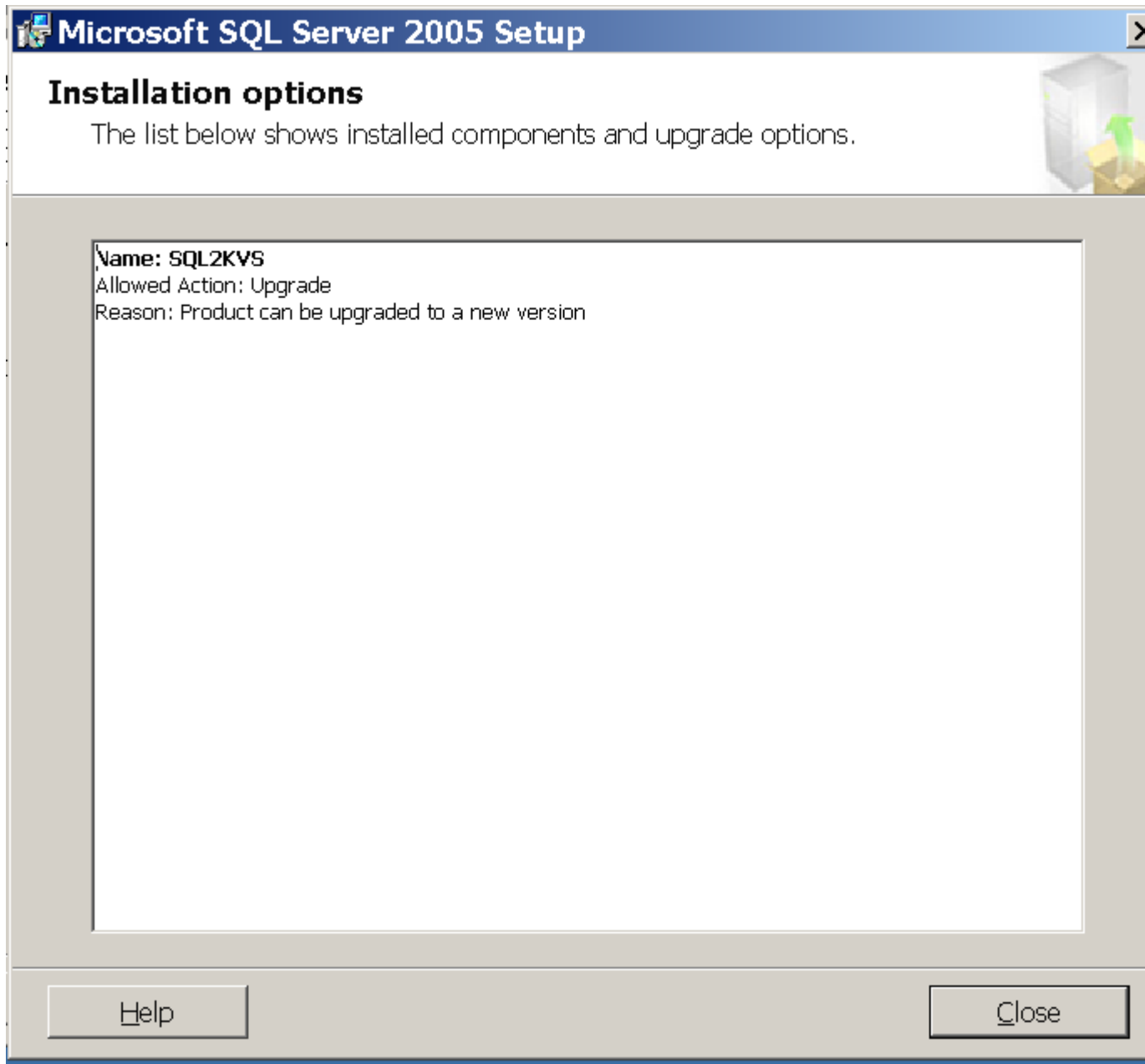
Figure 55. Installation Options page

37. On the Upgrade Logon Information page, click either Windows Authentication Mode or Mixed Mode of the account which will be used during the upgrade. Most installations will use Mixed Mode. Click **Next**.

Figure 56. Upgrade Logon Information page

The upgrade will now be verified and you will see a series of dialogs similar to the one below.
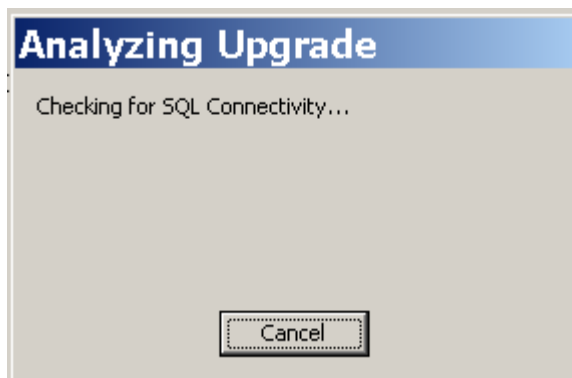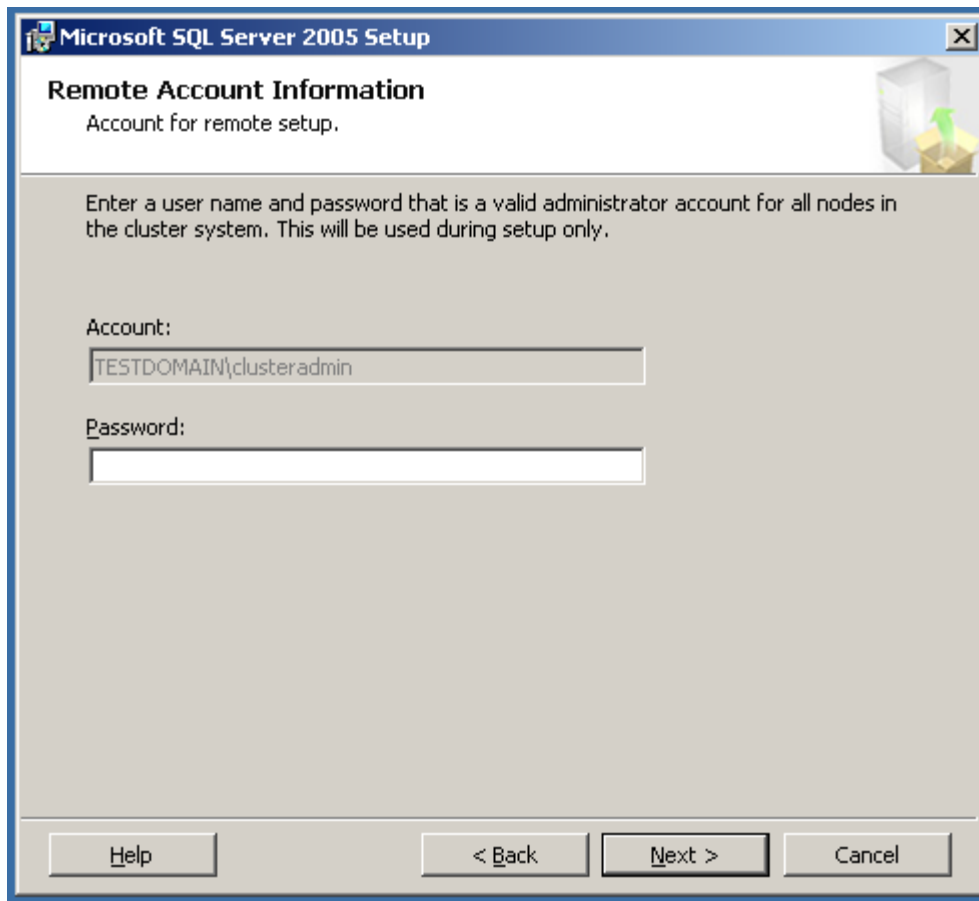

Figure 57. Analyzing Upgrade dialog box

38. On the Remote Account Information page, enter the password of the administrator whom you are logged in as. This administrator must have administrative privileges

on all nodes of the cluster. If the user you are currently logged in is not an administrator of the cluster nodes, cancel Setup, log out, and log in as the proper user. This user should not be the SQL Server administrative account. Click **Next**.


Figure 58. Remote Account Information page

39. The Service Account page configures the service accounts that SQL Server will use after it is installed. You will need to provide the password for the current Windows-based service account. Click **Next** when done.

Figure 59. Service Account page

40. On the Domain Groups for Clustered Services page, select the domain groups that will be used for the failover clustering instance. If you know the name of the groups, enter them in the appropriate text box in the format *DomainName\GroupName*.
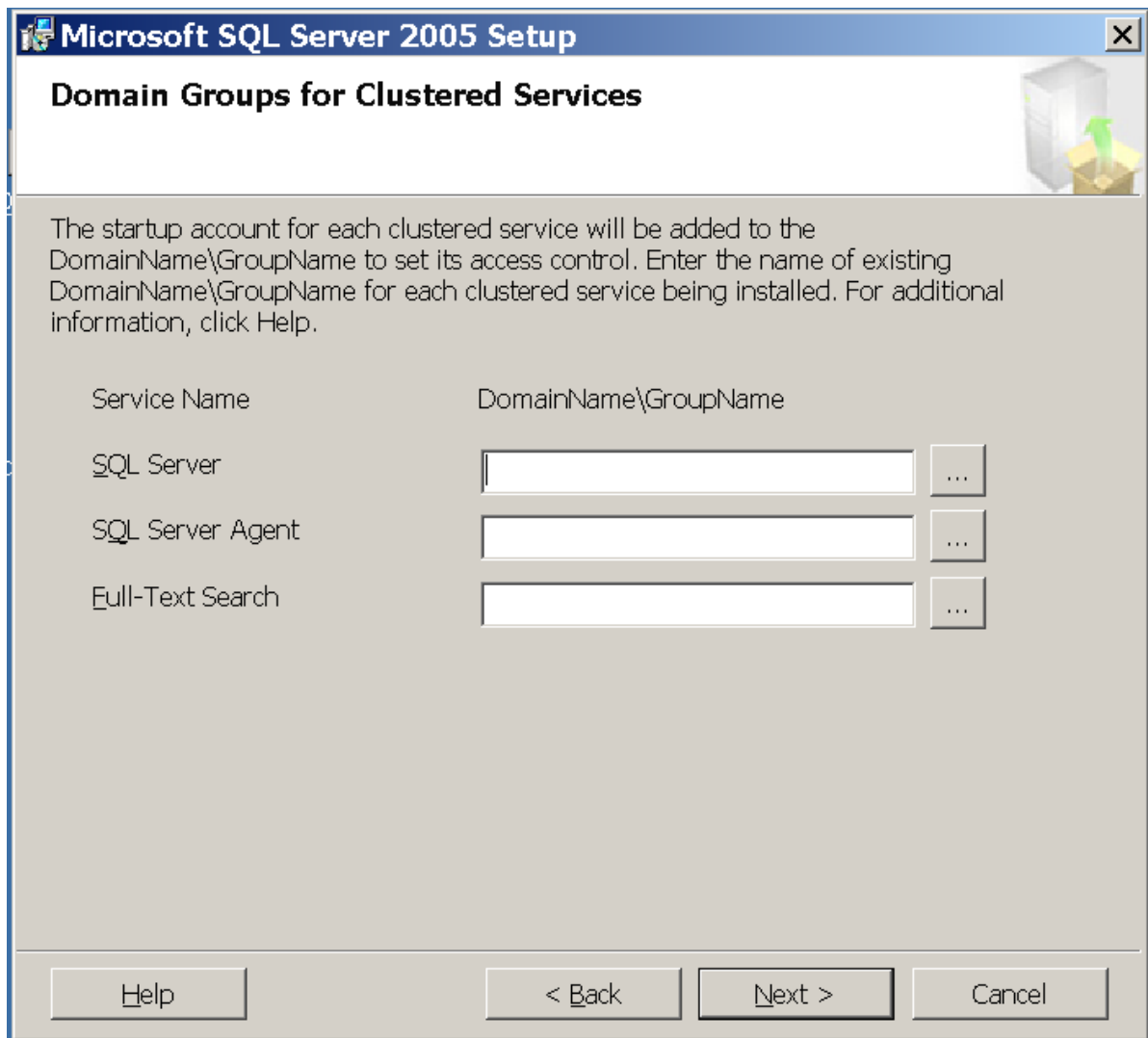
Figure 60. Domain Groups for Clustered Services page

If you do not know the name of the group, click the ellipsis button, and the Select Group page will be displayed. Select the domain to search, enter the name or part of the name, and click **Check Names**. Click **OK**.
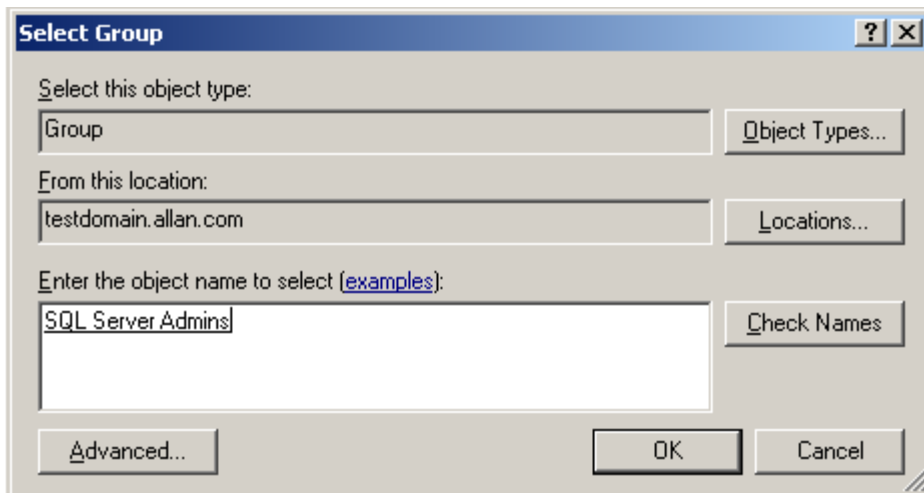
Figure 61. Select Group page

> **Important:** You must enter a domain group for each Service Name listed. The installation process cannot proceed if any entry is left blank. You may use the same group for all services listed if it has the proper privileges, but it is not recommended.

41. On the Ready to Install page, click **Next** to start the installation process.
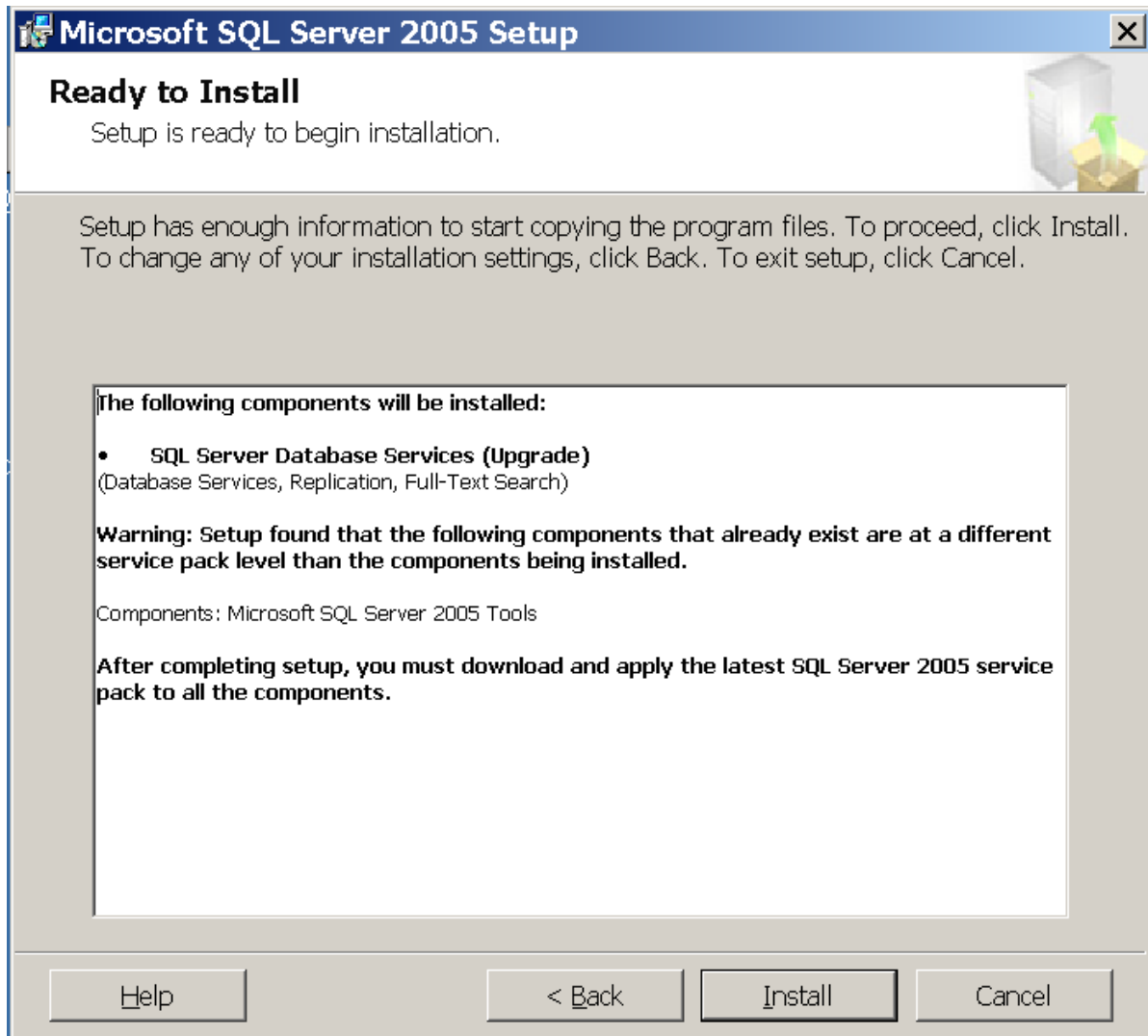
Figure 62. Ready to Install page

Below is the screen you will see as the installation options are configured to install. Depending on what options you selected in step 9, you may see additional items listed in your summary. During this time Setup will be accessing the other nodes to install any prerequisites.
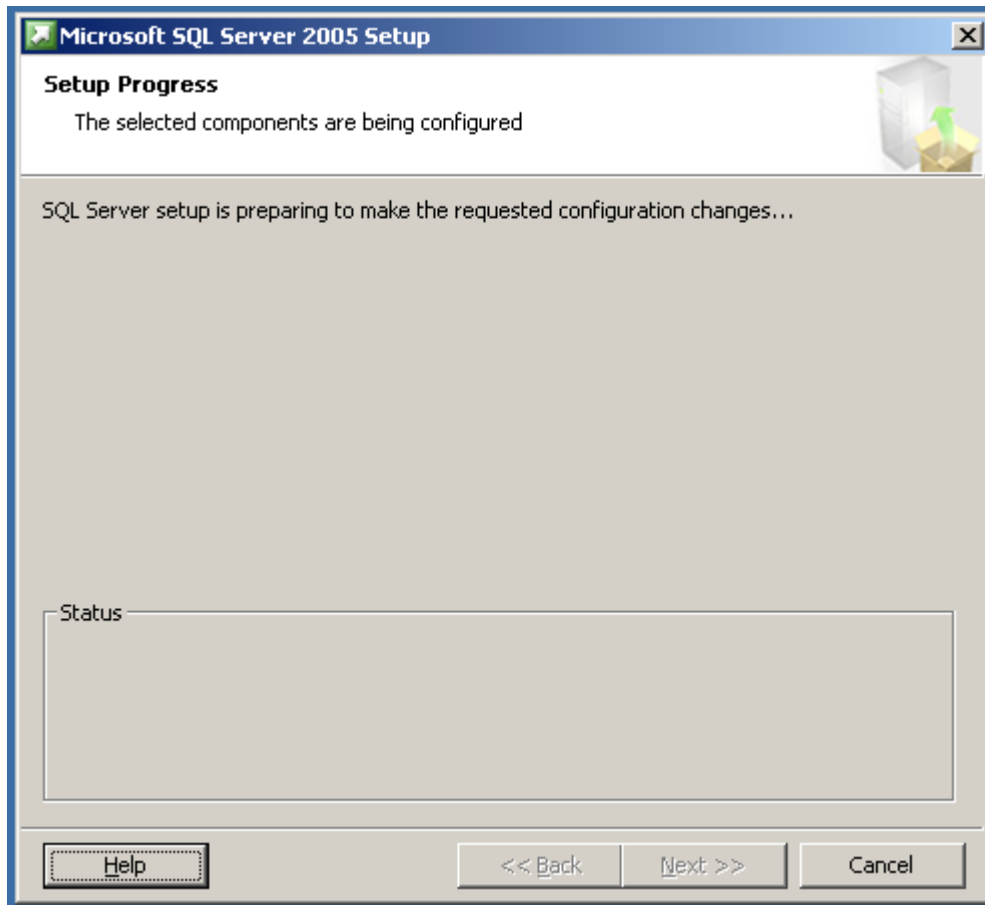
Figure 63. Setup Progress page

42. Once the installation process starts, you can monitor its progress. New to SQL Server 2005 Setup is the ability to see the progress of each node during clustered installs. To see another node, select it from the **Node** list.
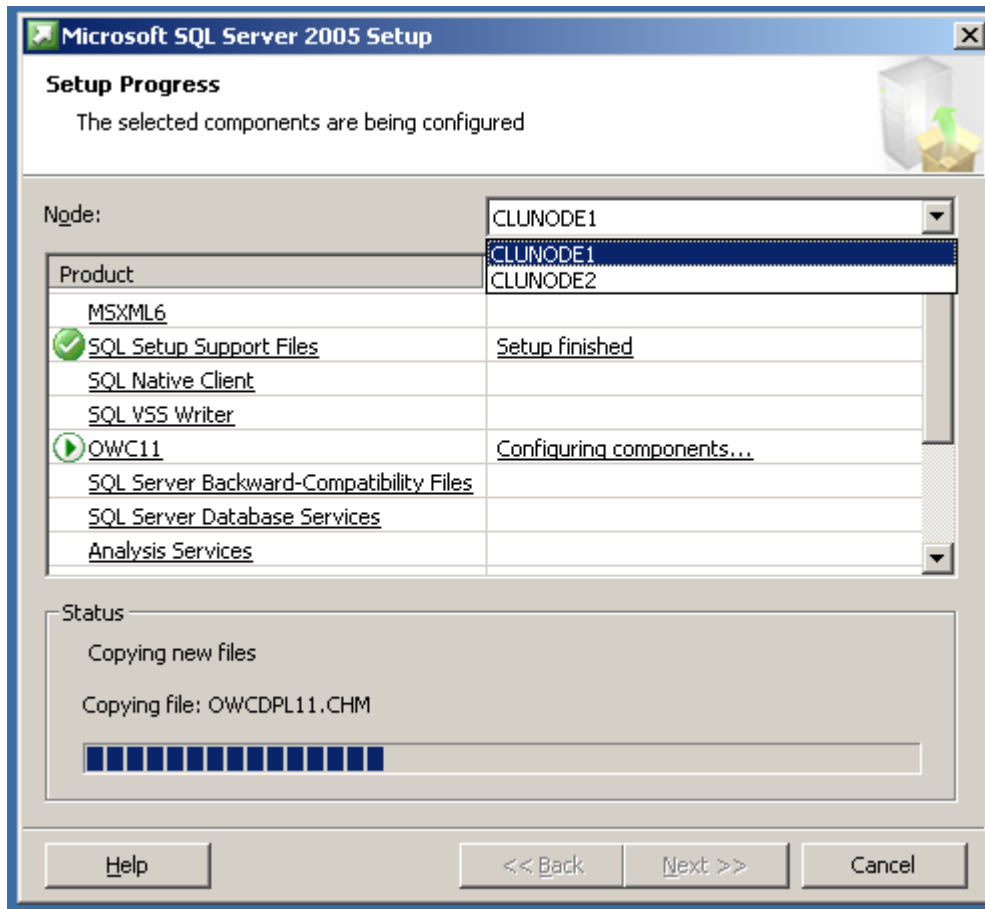
Figure 64. Setup Progress page

43. When setup is complete, the Setup Progress page will show all products with a status of Setup Finished. Click **Next**.
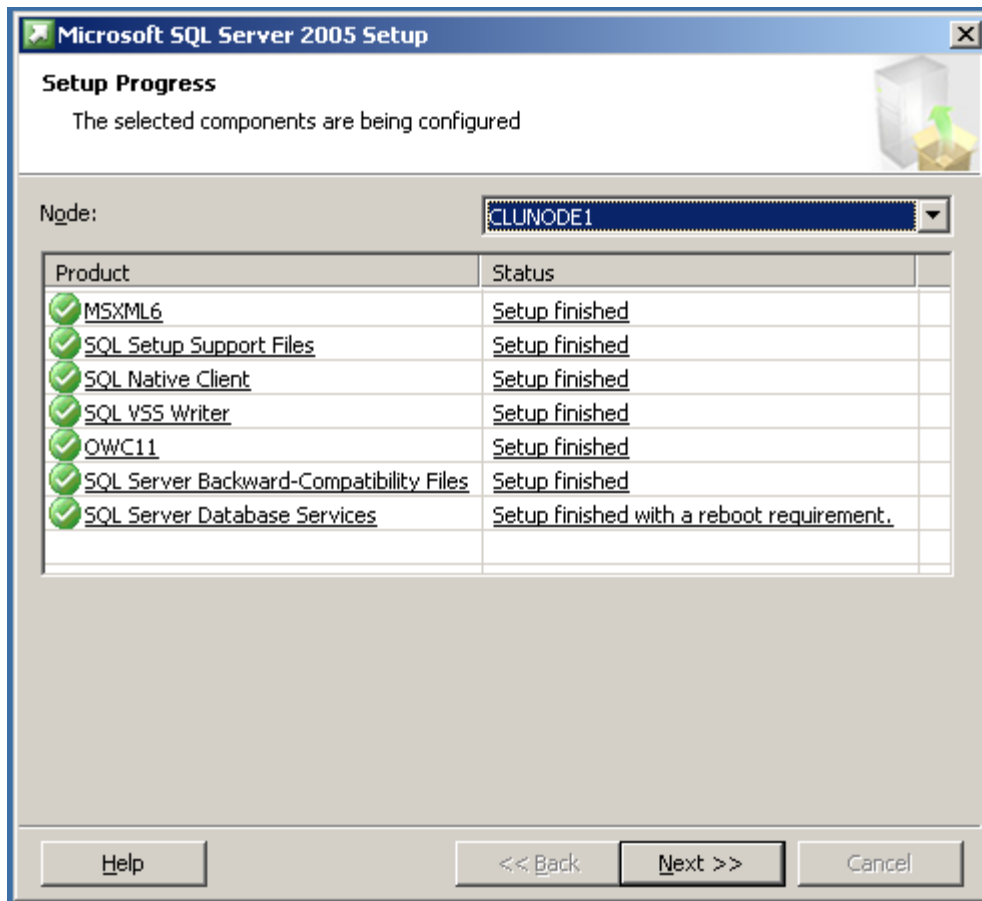
Figure 65. Setup Progress page

44. Read the information presented on the Completing Microsoft SQL Server 2005 Setup page, and click **Finish**.

Appendix C — Installing and Configuring Failover Clustering from a Command Prompt or an INI File

This section shows examples of running the specific installation and other configuration options related to failover clustering from the new command-prompt installation interface. All permutations will not be shown here, but the examples should show you how to create your own files if you need to do something different. You will need to customize these examples for your environment.

> **Note:** You may see slightly different behaviors between the command-prompt and the GUI-based Setup installation. The former installs are more flexible in certain instances.

For the .ini files, just cut and paste the text shown below to a file, and then run the file from the command prompt. You must have access to the SQL Server 2005 installation files, either on CD/DVD or a shared directory, as the install process uses the setup.exe that comes with the installer. Here is an example:

```
d:\servers\setup.exe /settings c:\sql2005clustersetup-asonly.ini /qb
```

You can also execute the commands directly at the command prompt. Each option listed in the .ini would be typed out minus the [Options]. For example:

```
start /wait d:\servers\setup.exe VS=SQL INSTALLVS=SQL_Engine
INSTANCENAME="INS2" ADMINPASSWORD=password ADDNODE="CLUNODE1" GROUP="SQL
Server 2005"
```

When executing from the command prompt, you can use either the `/qn` or `/qb` switches. `/qn` suppresses any dialogs, so if there is a problem, the only way to detect it would be to view the installation logs. By contrast, `/qb` displays some of the dialogs and errors as if you were running the GUI-based Setup. `/qb` is useful especially when running a script for the first time or debugging a script, but once the script is known to work, you may want to consider switching to using `/qn`.

When you execute from the command prompt, make sure to log into the node that currently owns the SQL Server resources, or log into a valid node of the failover cluster, and move the SQL Server resources to the node you are on via Cluster Administrator. You must log in as a user who has rights to administer the cluster nodes, otherwise these commands will fail.

> **Note:** The syntax below may differ slightly from the commands documented in How to: Install SQL Server 2005 from the Command Prompt. For a good starting resource, use the template.ini found on the SQL Server 2005 setup media, and check for any updates to the BOL topic in an updated version of Books Online.

> **Important:** All scripts must also have the PIDKEY option, which is the Product Identification Key. This can be found on your install media.

Install Analysis Services

```
[Options]

INSTALLSQLDIR="C:\Program Files\Microsoft SQL Server\"

INSTALLASDATADIR="M:\AS Data\"
```

```
ADDLOCAL="Analysis_Server,AnalysisDataFiles,Client_Components,Connectivity
"

INSTANCENAME="AS1"

ASACCOUNT="TESTDOMAIN\asadmin"

ASPASSWORD="password"


ASCOLLATION=SQL_Latin1_General_CP1_CI_AS


ERRORREPORTING=0

SQMREPORTING=0

ADMINPASSWORD=password

VS=SQL2K5AS

INSTALLVS="Analysis_Server"

IP="172.22.10.188,Public Network"

GROUP="Analysis Services"

ADDNODE="CLUNODE1,CLUNODE2"

ASCLUSTERGROUP="TESTDOMAIN\Analysis Services Admins"
```

Install SQL Server

```
[Options]

INSTALLSQLDIR="C:\Program Files\Microsoft SQL Server\"

INSTALLSQLDATADIR="I:\SQL Data\"

ADDLOCAL="SQL_Engine,SQL_Data_Files,SQL_Replication,SQL_FullText,Client_Co
mponents,Connectivity"

INSTANCENAME="INS1"

SQLBROWSERACCOUNT="TESTDOMAIN\sqladmin"

SQLBROWSERPASSWORD="password"

SQLACCOUNT="TESTDOMAIN\sqladmin"

SQLPASSWORD="password"

AGTACCOUNT="TESTDOMAIN\sqlagentadmin"

AGTPASSWORD="password"

SECURITYMODE=SQL

SAPWD=password


SQLCOLLATION=SQL_Latin1_General_CP1_CI_AS
```

```
ERRORREPORTING=0

SQMREPORTING=0

ADMINPASSWORD=password

VS=SQL2K5I1

INSTALLVS="SQL_Engine"

IP="172.22.10.187,Public Network"

GROUP="SQL Server 2005"

ADDNODE="CLUNODE1,CLUNODE2"

SQLCLUSTERGROUP="TESTDOMAIN\SQL Server Admins"

AGTCLUSTERGROUP="TESTDOMAIN\SQL Agent Admins"

FTSCLUSTERGROUP="TESTDOMAIN\FTS Admins"
```

Uninstall Analysis Services

```
[Options]

VS=SQL2K5AS

INSTANCENAME="AS1"

REMOVE="Analysis_Server"

ADMINPASSWORD="password"
```

Uninstall SQL Server

```
[Options]

VS=SQL2K5I1

INSTANCENAME="INS1"

REMOVE="SQL_Engine"

ADMINPASSWORD="password"
```

Install Client Tools Only

```
[Options]

INSTALLSQLDIR="C:\Program Files\Microsoft SQL Server\"

ADDLOCAL="SQL_DTS,Client_Components,Connectivity,SQL_Tools90,SQL_Warehouse
DevWorkbench,SQLXML,Tools_Legacy,SQL_Documentation,SQL_BooksOnline"

ERRORREPORTING=0

SQMREPORTING=0
```

Rebuild Master

```
[Options]

VS="SQL2K5I1"
```

```
INSTANCENAME="INS1"

REINSTALL="SQL_Engine"

REBUILDDATABASE=1

ADMINPASSWORD="password"

SQLACCOUNT="TESTDOMAIN\sqladmin"

SQLPASSWORD="password"

AGTACCOUNT="TESTDOMAIN\sqlagentadmin"

AGTPASSWORD="password"

; Only use if changing the collation – uncomment SQLCOLLATION

; and add a valid collation

;SQLCOLLATION=SQL_Latin1_General_CP1_CI_AS
```

Appendix D – Installing a SQL Server 2005 Service Pack in a Cluster

1. After downloading the service pack installer to the node that owns the SQL Server instance you are updating, double click the executable (for example, SQLServer2005SP1-KB913090-x86-ENU.exe) to start the installation process. While it is starting, you will see a dialog similar to the one below. When you see the Welcome page, click **Next**.
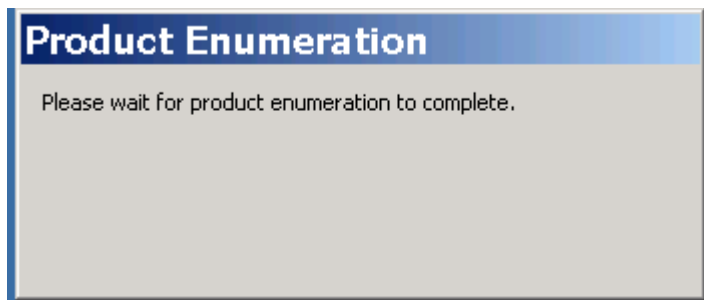

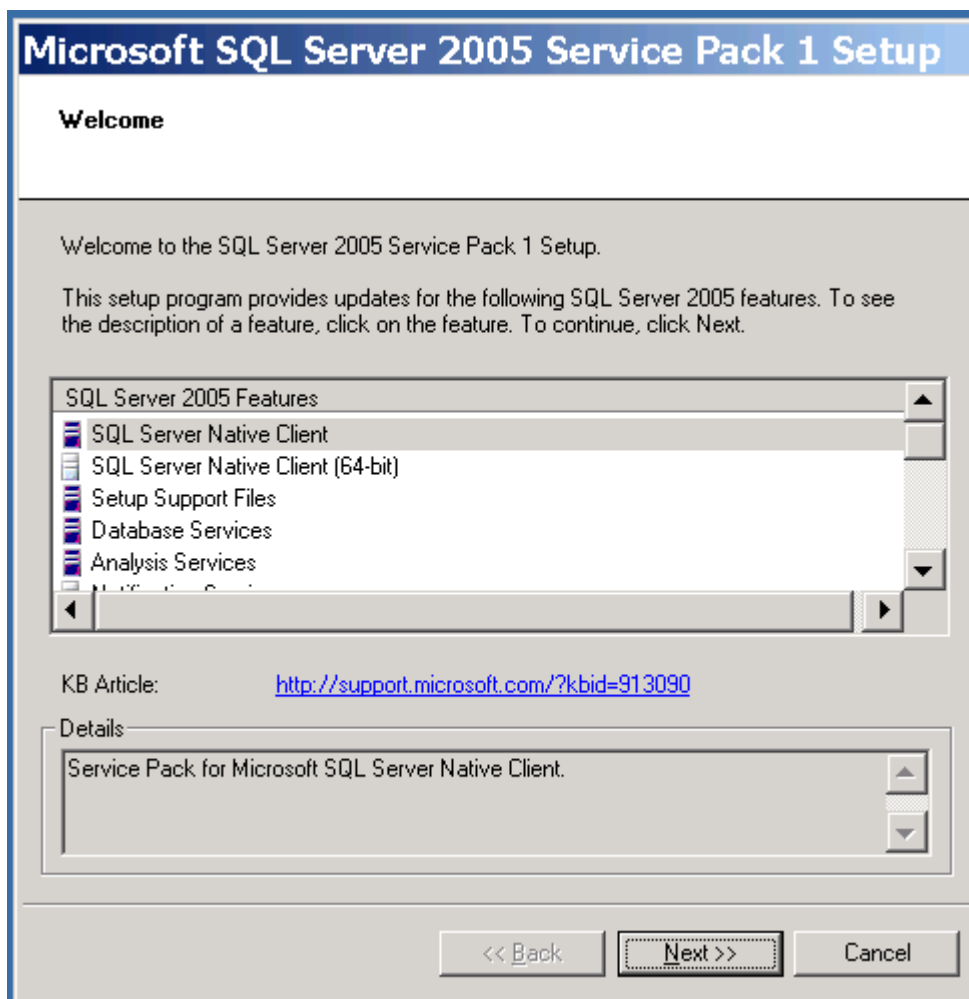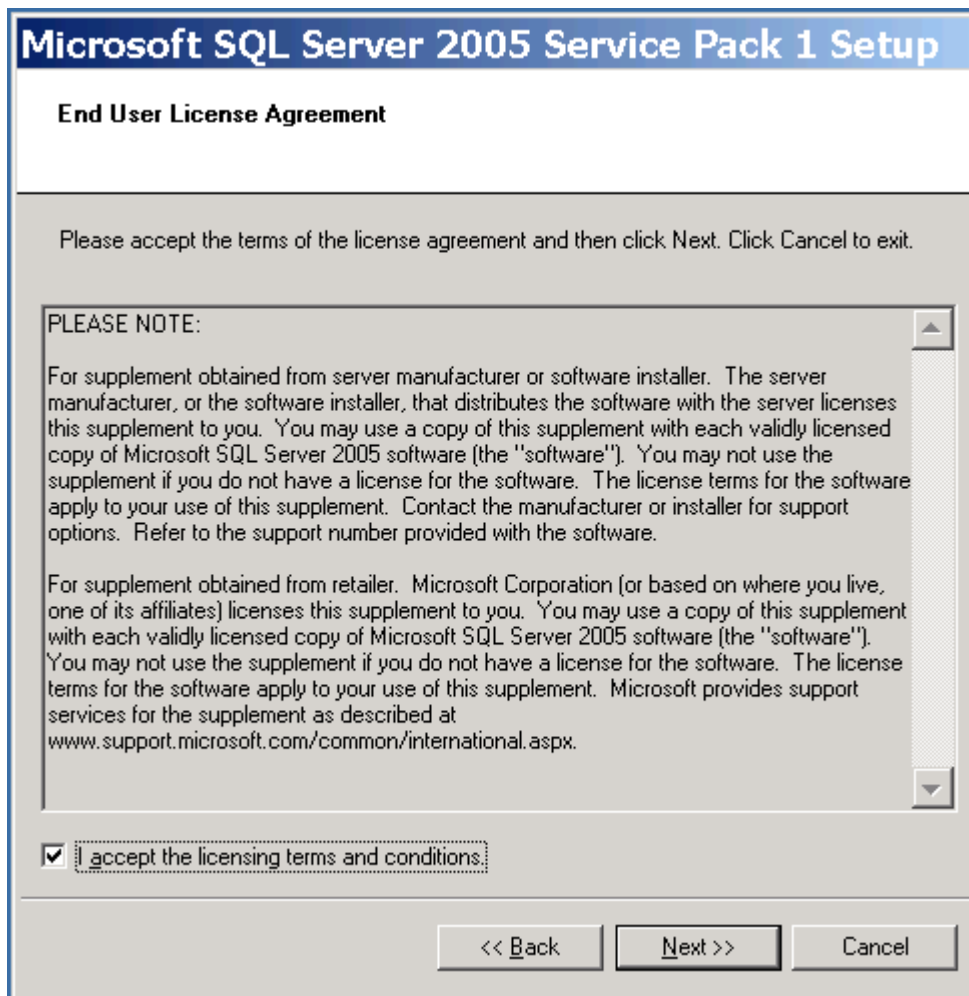**Figure 66. Product Enumeration dialog box**

Figure 67. Welcome page

2. Select **I accept the licensing terms and conditions** after reading the End User License Agreement and click **Next**.

Figure 68. End User License Agreement

3. The Feature Selection page will show the components that are available for upgrading. You can deselect some of the features, but note that if you select certain features, the page will indicate whether the update is required or if all of the features related to that instance of the relational engine or Analysis Services must be upgraded at the same time. Click **Next** when ready to proceed.
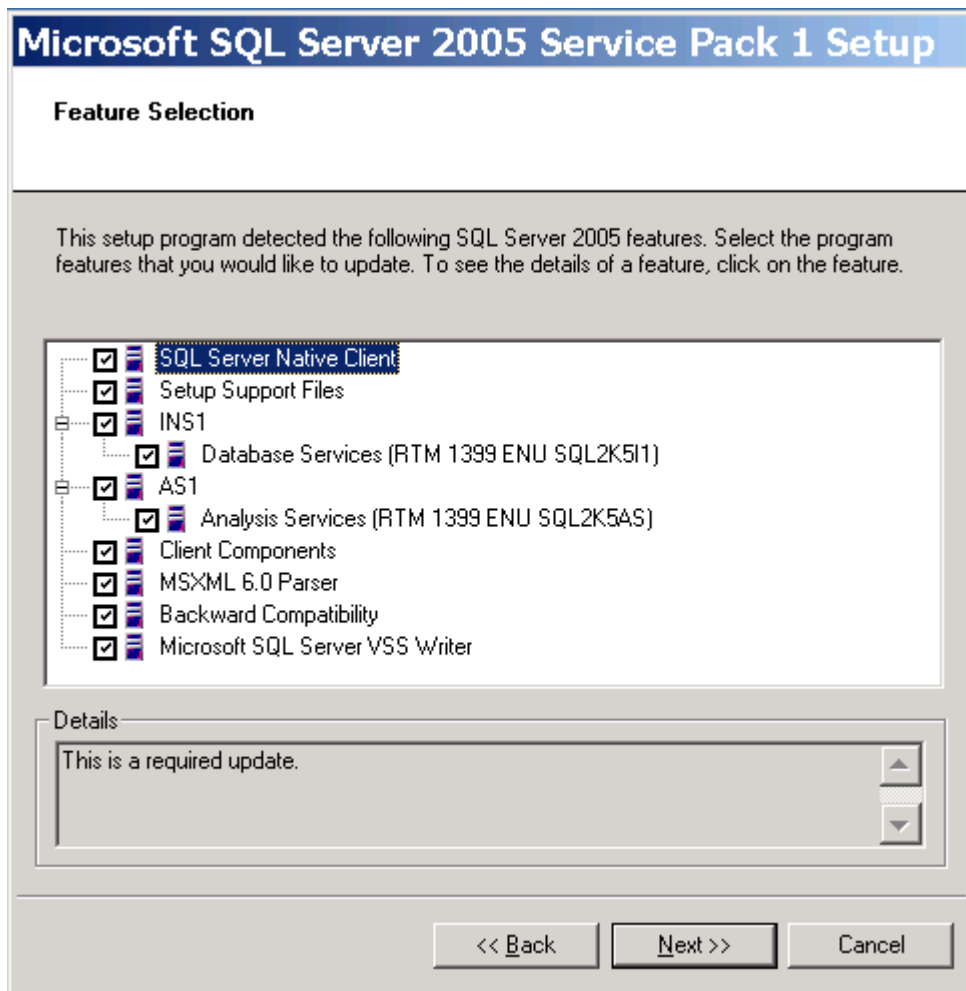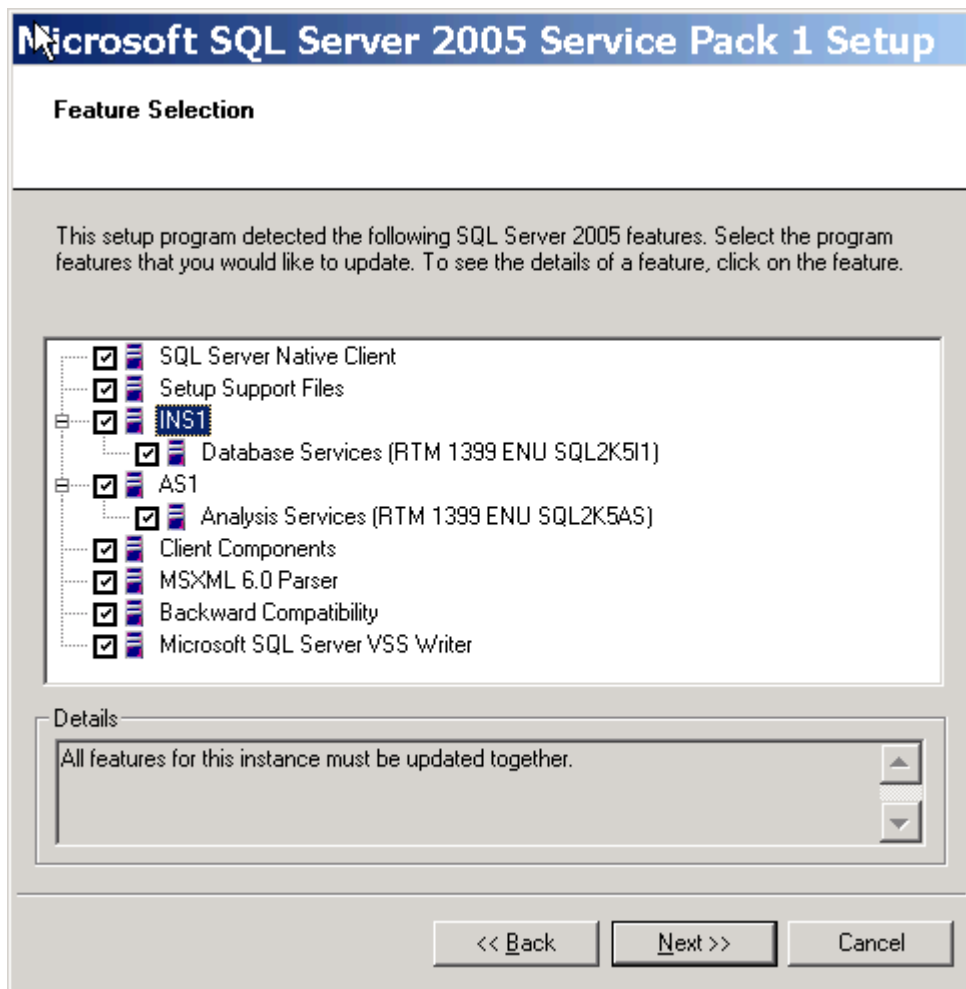
Figure 69. Feature Selection page

Figure 70. Feature Selection page

4. On the Authentication Mode page, select the method by which the service pack will connect to SQL Server. If you select SQL Server Authentication, you must enter the **sa** password. Click **Next**.

Figure 71. Authentication Mode page

5. On the Remote User Account page, enter the name of the cluster administrator account that has the privileges to write files to each node of the cluster in **Username**, the account password in **Password**, and the account domain in **Domain**.

Figure 72. Remote User Account page
Click **Next**.

6. On the Ready to Install page, click **Install**.
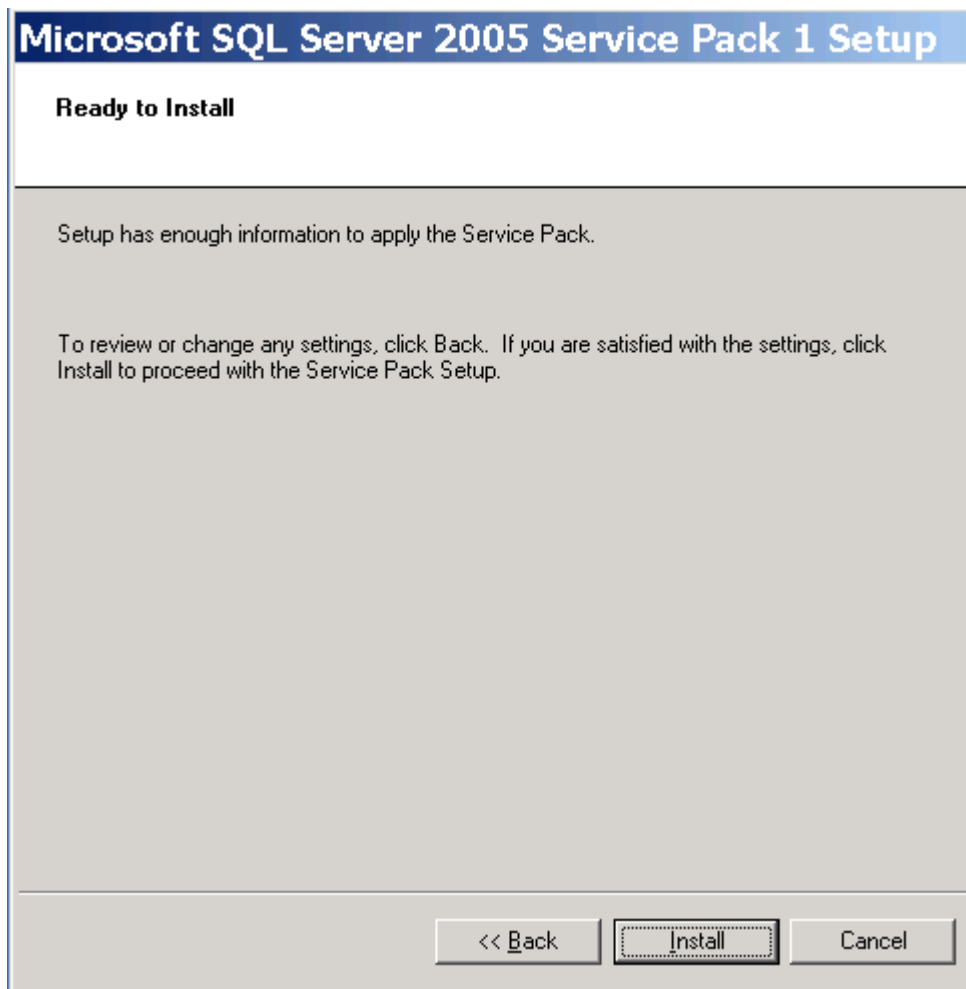
Figure 73. Ready to Install page

During the installation process, the status of each component will appear on the Installation Progress page. A checkmark will appear when that feature is completely installed. In the **Status** column, the message "Reboot Required" will appear if that feature will cause all nodes to be rebooted after the installation is complete.
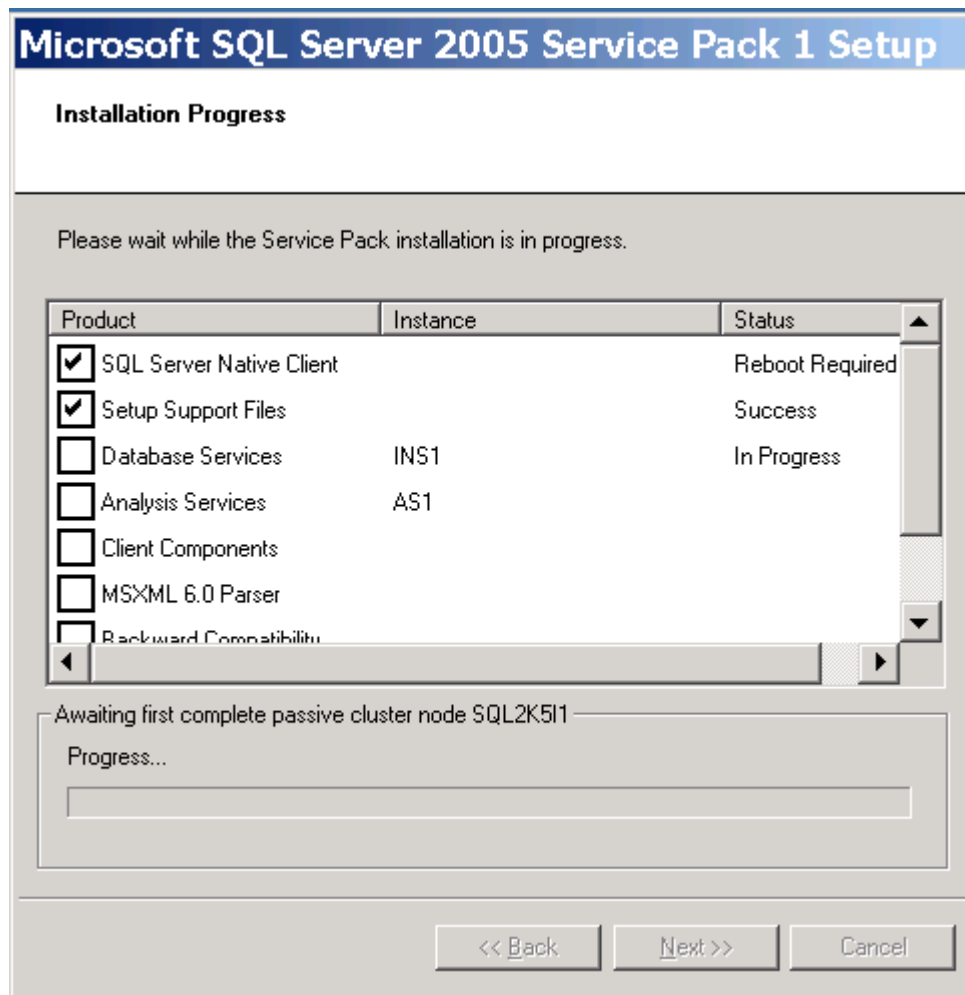
Figure 74. Installation Progress page

The installer also detects whether the installation of a program before the service pack left pending file operations that may have required a reboot.
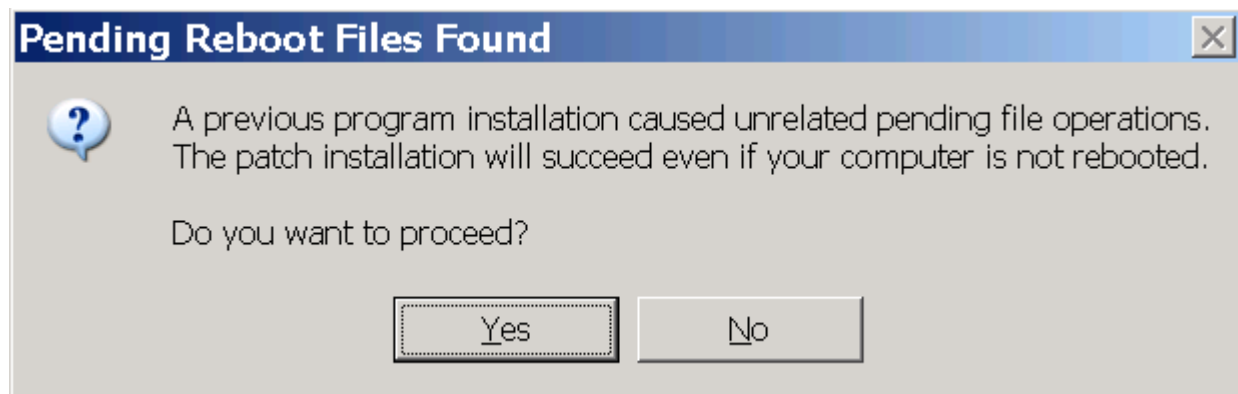
Figure 75. Pending Reboot Files Found dialog box

The installer will detect files that need to be upgraded but may be locked (for example, if SQL Server Configuration Manager was left open on the server).
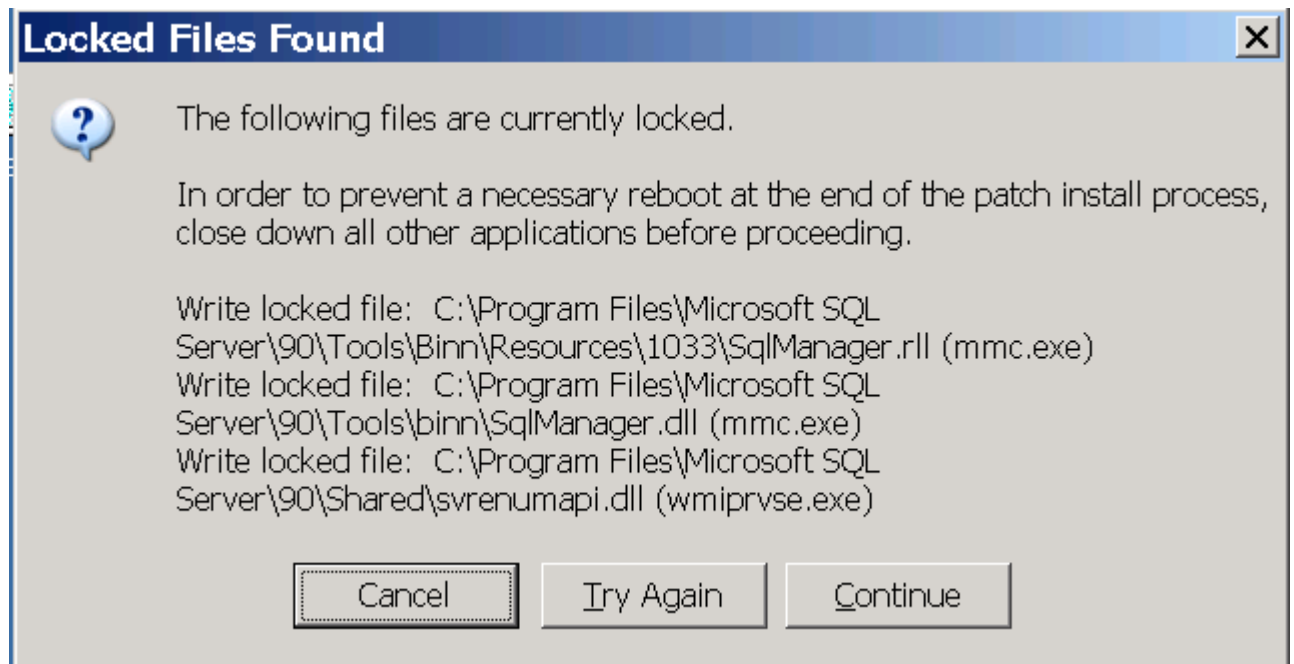
Figure 76. Locked Files Found dialog box

7. When the install process is complete, a message box will be displayed if a reboot is required. If so, click **OK**.
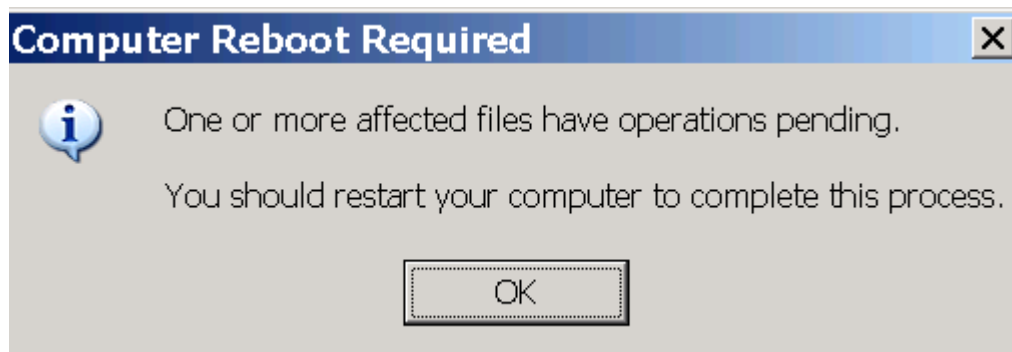
Figure 77. Computer Reboot Required dialog box

8. Click **Next**.
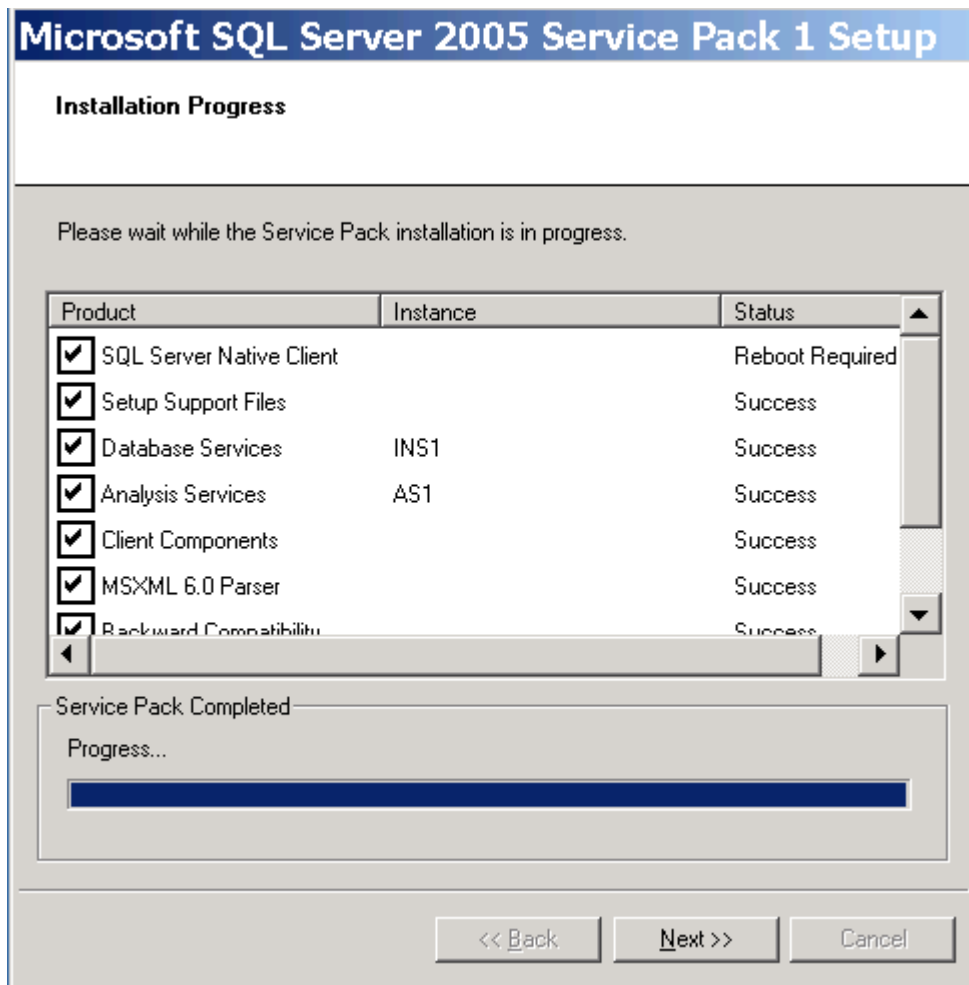
Figure 78. Installation Progress page

9. On the Installation Complete page, click **Finish**.
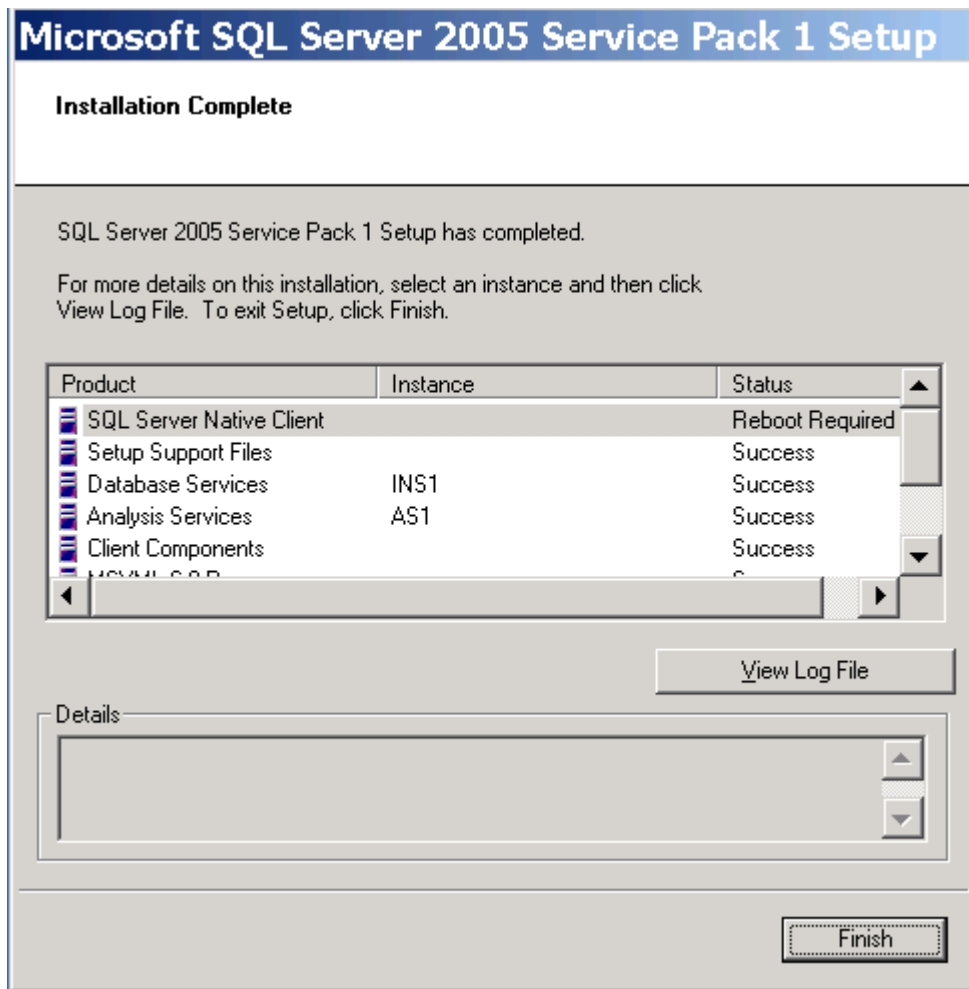
Figure 79. Installation Complete page

10. If required, reboot all nodes that are part of the definition for the instance that has been upgraded.

11. Once the nodes are rebooted, if the SQL Server management tools are installed on the other nodes, re-run the service pack executable on those nodes to upgrade the tools.

Appendix E – SQL Server 2005 Directory and Registry Structures

The on-disk structure of SQL Server 2005 differs from that of SQL Server 2000. SQL Server 2000 is installed into a directory bearing the name of the instance being installed and its settings are contained in a registry key bearing its name. To ensure that things are not disturbed in a side-by-side scenario with SQL Server 2000 and SQL Server 2005, not all binaries can be installed into the same directory as the version that is being upgraded.

Because SQL Server forces the new installation into a different structure than the one used with SQL Server 2000, SQL Server 2005 provides better support for upgrades. SQL Server Setup uses an instance GUID instead of an instance name in the directory and registry paths. The instance GUID is a number that is the next in serial order to the highest installed instance GUID on the machine. The instance GUID is generated by SQL Server Setup during installation. If gaps occur in the GUID sequence due to uninstalls, IDs will not be generated to fill them; the GUID generated by Setup will always be the next higher in the sequence.

This approach may appear a bit confusing since each subdirectory is enumerated instead of named, but the advantage is that in the "shared" section of the SQL Server 2005 registry entries, Setup maps the actual directory names and registry paths to the instance names, so SQL Server knows what to access. This is why SQL Server 2005 now has the ability to rename clustered instances .

Instance-Specific Directory Structure

The SQL Server 2005 directory structure is as follows:

- *%drive%*\Program files\Microsoft SQL Server\90*instancename*MSSQL.*<Instance GUID>*.
- \Binn — SQL Server binaries directory, basic contents remain the same as with existing SQL Server 2000 installations.
- \Data — directory, contents remain the same as with existing SQL Server 2000 installations.

For more information, see [File Locations for Named and Default Instances of SQL Server 2005](#).

Tools Directory Structure

- "*%drive%*\Program Files\Microsoft SQL Server\90\shared

Registry Structure

Registry format for instance-specific information and the common shared information:

- HKLM\SYSTEM\Software\Microsoft\SQL Server\90*instancename GUID*
  and HKLM\SYSTEM\Software\Microsoft\SQL Server\90\shared.

Appendix F – Failover Clustering Dependencies

The table below lists the names of the SQL Server resources, what they were in 2000 for comparison, and what dependencies they require.

SQL IP Address*X* (*VSName*)	SQL IP Address *X* (*VSName*)	None
SQL Network Name(VSName)	SQL Network Name (*VSName*)	SQL IP Address
SQL Server or SQL Server(*InstanceName*)	SQL Server or SQL Server (*InstanceName*)	Disk Resource(s), SQL Network Name
SQL Server Agent or SQL Server Agent(*InstanceName*)	SQL Server Agent or SQL Server Agent (*InstanceName*)	SQL Server
SQL Server Fulltext or SQL Server Fulltext(*InstanceName*)	SQL Server Fulltext or SQL Server Fulltext (*InstanceName*)	Disk Resource(s)
N/A	Analysis Services or Analysis Services (*InstanceName*)	Disk Resource(s), SQL Network Name

X is a number, *VSName* is the name of the server cluster virtual server you assigned in Setup, and *InstanceName* is the name of the named instance assigned during setup.

Verbose logging on the SQL Server resource DLLs can be set using the command-prompt functionality of Cluster Administrator.

Note: The SQL Server Full-Text resource type in SQL Server 2005 has changed from SQL Server 2000. It is now a Generic Resource. Each instance installed onto a cluster gets its own Full-Text resource instead of sharing the underlying Full-Text service as SQL Server 2000 failover clustering did. The Full-Text resource is also now known as the msftesql service.

Note: The SQL Server 2005 DLL defaults to a 180-second time-out. This was first implemented in SQL Server 2000 Service Pack 4; before that, the time-out defaulted to 60 seconds.

Appendix G – Cluster Terminology Overview

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:

Cluster Nodes

A node is a physical server configured as part of a Windows server cluster.

Heartbeat

The heartbeat is a private network set up between the nodes of the cluster that is checked 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 configuration 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 when two computers have concurrent access to the same drive.

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 the quorum drive becomes corrupted or damaged, the cluster installation also becomes corrupted or damaged. The quorum drive is used in a "shared quorum" configuration; for a "local quorum" (or Majority Node Set) configuration, a local disk is used on each node.

Operating System Components

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

Cluster Name

The name that Windows Server 2003 external connections use to refer to the cluster itself, not the SQL Server virtual server. Individual cluster 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 on 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 like SQL Server 2005. Conceptually, a cluster group is a folder on your hard drive that contains related information.

Failover Clustered Instance

Since the release of Microsoft Virtual Server 2005, the term "Virtual Server" is no longer used in the context of failover clustering. The term "failover clustered instance" (FCI) is now used to eliminate this confusion.

Understanding the concept of a failover clustered instance, previously referred to as a virtual server, is key to understanding failover clustering. To a client or application, a virtual server is the server name or IP address 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 instance of SQL Server is known as a SQL Server virtual server.

Additional Resources

For more information about SQL Server 2000 failover clustering, see the [SQL Server 2000 Failover Clustering](#) white paper.

For more information about SQL Server 2005 failover clustering, see the following resources:

- [Microsoft SQL Server Web site](#)
- [SQL Server 2005 Books Online](#)
- [Windows Server 2003 Technology Center Home](#)

Did this paper help you? Please give us your feedback. On a scale of 1 (poor) to 5 (excellent), [how would you rate this paper](#)?