# logo-sql08.gif

SQL Server 2008 Failover Clustering

SQL Server Technical Article

**Writers:** Mike Weiner, Paul Burpo, Max Verun, Joseph Sack, Justin Erickson

**Contributors:** Sanjay Mishra, Jason Wu, Uttam Parui

**Technical Reviewers:** Prem Mehra, James Podgorski, David Whitney, Richard Tkachuk, Sethu Kalavakur, Cindy Gross, Neal Graves, Farzan Ratistari, Ayad Shammout (Caregroup Healthcare Systems), David P. Smith (ServiceU Corporation)

**Published:** June 2009

**Applies to:** SQL Server 2008

**Summary:** This white paper complements the existing documentation on planning, implementation, and administration of a SQL Server 2008 failover cluster, which can found in Microsoft SQL Server 2008 Books Online. There are links to relevant existing content throughout the paper, which is intended primarily for a technical audience. This white paper covers failover cluster architecture and concepts for Windows Server and SQL Server 2008; installation of a SQL Server 2008 failover cluster; upgrades and updates to SQL Server 2008 failover clustering; and maintenance and administration of SQL Server 2008.

Copyright

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

© 2009 Microsoft Corporation. All rights reserved.

Microsoft, Active Directory, Excel, Hyper-V, SQL Server, Win32, Windows, Windows Server, and Windows Vista are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

Contents

[Introduction 5](#_Toc233413068)

[Overview of Windows Server Failover Clustering 5](#_Toc233413069)

[Topology and Architecture of a SQL Server Cluster 8](#_Toc233413070)

[Failover Cluster Resource Group 9](#_Toc233413071)

[Health Detection 10](#_Toc233413072)

[Resource and Resource Group Properties 11](#_Toc233413073)

[Resource Properties 11](#_Toc233413074)

[Resource Group Properties 13](#_Toc233413075)

[Common SQL Server Failover Deployments 14](#_Toc233413076)

[Single Failover Cluster Instance 14](#_Toc233413077)

[Multi-Instance Failover Cluster 15](#_Toc233413078)

[Multi-Site Failover Cluster 15](#_Toc233413079)

[Guest Failover Cluster 17](#_Toc233413080)

[Multiple SQL Server Versions Within the Same Windows Server Cluster (SQL Server 2000 and 2005 with SQL Server 2008) 18](#_Toc233413081)

[Advantages of SQL Server 2008 on Windows Server 2008 Clustering 19](#_Toc233413082)

[SQL Server Installation and Setup 20](#_Toc233413083)

[Installation Options for a SQL Server Failover Cluster 21](#_Toc233413084)

[Integrated Installation 21](#_Toc233413085)

[Advanced/Enterprise Installation 21](#_Toc233413086)

[Adding a Node to a Failover Instance 22](#_Toc233413087)

[Removing a Node from a Failover Instance 22](#_Toc233413088)

[Uninstalling a SQL Server Instance 23](#_Toc233413089)

[Unattended Installation of SQL Server 2008 Failover Clustering 23](#_Toc233413090)

[Using a Configuration File 26](#_Toc233413091)

[Post-Installation Considerations 29](#_Toc233413092)

[Adding Additional Cluster Disks for SQL Server 29](#_Toc233413093)

[Verify SQL Server Resource Failover 29](#_Toc233413094)

[SQL Server Management Tools and Client Installation 29](#_Toc233413095)

[Using the Command-Line Setup 30](#_Toc233413096)

[Removing the SQL Server Client Tools 30](#_Toc233413097)

[Considerations for Installing Other Components in a Failover Cluster 31](#_Toc233413098)

[Analysis Services 31](#_Toc233413099)

[Reporting Services 32](#_Toc233413100)

[SQL Server Integration Services 32](#_Toc233413101)

[Microsoft Distributed Transaction Coordinator 32](#_Toc233413102)

[SQL Server Upgrade from Previous Versions 36](#_Toc233413103)

[Reducing Downtime by Using Rolling Updates and Upgrades 37](#_Toc233413104)

[Rolling Upgrades 37](#_Toc233413105)

[Rolling Updates 43](#_Toc233413106)

[Administration and Maintenance of a SQL Server 2008 Failover Cluster 49](#_Toc233413107)

[Changing the Network Name of a SQL Server Failover Cluster 49](#_Toc233413108)

[Changing the IP Address of a SQL Server Failover Cluster 50](#_Toc233413109)

[Changing the Service Accounts for a SQL Server Failover Cluster 50](#_Toc233413110)

[Managing SQL Server Resources by Using SQL Server Management Studio and Cluster Administrator 50](#_Toc233413111)

[Rebuilding System Databases in a SQL Server Failover Cluster 51](#_Toc233413112)

[Implementing Trace Flags 51](#_Toc233413113)

[Troubleshooting SQL Server Failover Clusters 52](#_Toc233413114)

[Determining the Nature of the Failure 52](#_Toc233413115)

[Root Cause Analysis 53](#_Toc233413116)

[Hardware 53](#_Toc233413117)

[Operating System and Network 53](#_Toc233413118)

[Security 54](#_Toc233413119)

[Windows Server Failover Clustering 54](#_Toc233413120)

[SQL Server 55](#_Toc233413121)

[Further Troubleshooting Techniques 56](#_Toc233413122)

[Reading and Extracting Event Logs and Cluster Logs 56](#_Toc233413123)

[Comparing Registries Across Nodes 57](#_Toc233413124)

[Manually Changing a Checkpointed Key 57](#_Toc233413125)

[Bringing SQL Server Online Outside Cluster Services 58](#_Toc233413126)

[Properties for the SQL Server Resource 59](#_Toc233413127)

[Best Practices to Prevent Unnecessary Outages 60](#_Toc233413128)

[Evaluate Resource Dependencies 60](#_Toc233413129)

[Evaluate Resource Failover Policies 61](#_Toc233413130)

[Connectivity Configuration 62](#_Toc233413131)

[Appendix A: Installing a SQL Server 2008 Failover Cluster Using Integrated Installation with Add Node Operation 62](#_Toc233413132)

[Appendix B: Installing a SQL Server 2008 Failover Cluster Using Advanced/Enterprise Installation 93](#_Toc233413133)

[Advanced/Enterprise Failover Cluster Install Step 1: Prepare 93](#_Toc233413134)

[Advanced/Enterprise Failover Cluster Installation Step 2: Complete 115](#_Toc233413135)

[Appendix C: Add Node 132](#_Toc233413136)

[To Add a Node to an Existing SQL Server 2008 Failover Cluster 132](#_Toc233413137)

[Appendix D: Removing a Node 147](#_Toc233413138)

[To Remove a Node from an Existing SQL Server 2008 Failover Cluster 147](#_Toc233413139)

[Appendix E: SQL Server 2008 Failover Cluster Setup - Troubleshooting Error Conditions 154](#_Toc233413140)

[Error Messages and Troubleshooting Guidance 156](#_Toc233413141)

[Troubleshooting Error Messages in “\*ErrorLog” Files 174](#_Toc233413142)

[Conclusion 176](#_Toc233413143)

# Introduction

Microsoft® SQL Server® 2008 provides a solution for mission-critical applications that require the highest levels of availability. SQL Server 2008 failover clustering is part of the SQL Server high-availability technology toolset and is designed to help businesses meet their availability and uptime goals. This provides protection against both planned as well as unplanned downtime. When a server on one of the node fails SQL Server can continue serving request through other node(s).

SQL Server 2008 includes several changes to the implementation of SQL Server failover clustering, including an entirely new setup process and support for up to 16 nodes. This white paper discusses several new and existing concepts for SQL Server 2008 failover clustering, such as architecture, implementation, administration, and troubleshooting.

# Overview of Windows Server Failover Clustering

SQL Server 2008 failover clustering is built upon an established Windows Server® failover cluster. A Windows Server failover cluster aims to provide high availability for services or applications that run within the failover cluster. It contains a group of independent servers that work together to increase the availability of applications and services. Failover clustering can protect against hardware and software failures by *failing over* resources from one server (or *cluster node*) to another as required. Failover is the process of taking a clustered service or application offline on one node and bringing it back online on another node. This process is typically transparent to the users, who should experience a minimal disruption of service when a failover occurs.

Other types of cluster configurations such as Windows® Compute Cluster Server 2003, which Windows® HPC Server 2008 succeeded, and Windows Network Load Balancing (NLB) clusters exist. However, all of them have different implementations and are not the Windows Server failover clustering for which SQL Server 2008 failover clustering is designed. A Windows failover cluster architecture provides redundancy and the ability to detect certain failures. It compensates for these failures by moving highly available resources over to redundant working components. This architecture is not suitable for scenarios such as load balancing and its implementations differ from the HPC and NLB cluster solutions.

A failover cluster contains one or more clustered servers (called nodes) and a configuration of shared cluster disks that are set up for use within the cluster. The failover cluster also enlists at least two networks for communication, at least one public and one private. The private network should be configured for internal cluster communication only. The public network, however, is primarily for clients to connect to the applications that are running within the cluster. The cluster nodes communicate with one another over the private network by using *heartbeats* (periodic health detection signals that enable them to assess the status of other nodes in the cluster).

Other failover cluster components (using definitions from the [Failover Cluster Design Guide](http://technet.microsoft.com/en-us/library/dd197457(WS.10).aspx), http://technet.microsoft.com/en-us/library/dd197457(WS.10).aspx) are:

* **Cluster service**. The essential software component that controls all aspects of server cluster or failover cluster operations. Each node in a failover cluster or server cluster runs one instance of the Cluster service.
* **Quorum**. The number of elements that must be online for a given failover cluster to continue running. The relevant elements in this context are nodes, a witness/quorum disk, or, in some cases, a witness file share. Each element included in the quorum, except a witness file share, contains a copy of the cluster configuration. The Cluster service works to keep all copies of the cluster configuration synchronized at all times.

Within the cluster, disks, IP addresses, network names, and applications such as SQL Server are resources that can be established as highly available and provide failover capabilities. These resources are designed to be cluster-aware through a resource dynamic-link library (DLL). The DLL contains the **resource-**specific code that is necessary to provide high availability for one or more **resource** types. Each resource type exposes two types of health detection checks:

* **LooksAlive** is a lightweight, more frequent check.
* **IsAlive** is run less frequently to do a deeper check that the service is available.

Resources that are related to a specific application instance are installed into a Windows Server failover cluster group, which is known as a resource group.

**Note:** In the Failover Cluster Management tool in Windows Server 2008, the **Resource Group** is called **Services and Applications**. These terms are used interchangeably throughout this document.

The resource group is the unit of failover, which means that all resources within the same group will fail over as one. At any given time, only one node in the cluster can own each resource group and the resources that it contains. See the “Topology and Architecture of a SQL Server Cluster” section below for more information about resources and resource groups for SQL Server.

SQL Server 2008 failover clustering can be configured on Windows Server 2003 and Windows Server 2008. Many of the references and definitions in this paper focus on Windows Server 2008. Some specific changes to Windows Server 2008 that relate to SQL Server are:

* **The cluster validation wizard**. In Windows Server 2003, the cluster hardware solution was validated manually via the [Windows Server Catalog](http://www.windowsservercatalog.com/results.aspx?&chtext=&cstext=&csttext=&chbtext=&bCatID=1291&cpID=0&avc=10&ava=0&avq=0&OR=1&PGS=25&ready=0). In Windows Server 2008, the cluster validation wizard provides this functionality and validates that the cluster environment is compatible with Windows Server 2008 failover clustering prior to setup. For a complete discussion on these tools and validation see “[Failover Cluster Step-by-Step Guide: Validating Hardware for a Failover Cluster](http://technet.microsoft.com/en-us/library/cc732035.aspx)” (http://technet.microsoft.com/en-us/library/cc732035.aspx)
* **Service security IDs (SIDs)**. If you are creating a new SQL Server 2008 failover cluster on Windows Server 2008, you can now bypass the use of domain groups by designating service SIDs during the installation. Service SID functionality was introduced in Windows Vista and Windows Server 2008, and allows the provisioning of access control lists (ACLs) to server resources and permissions directly to a Windows Server service. During the installation of a SQL Server failover cluster, in the **Cluster Security Policy** dialog box, you still have the option to use domain groups. However, for SQL Server 2008 on Windows Server 2008, the recommended choice is to select **Use service SIDs**. This enables you to bypass provisioning of domain groups and associated service account membership additions prior to installation.
* **Four quorum modes**. Windows Server 2008 failover clustering now supports four quorum modes. To provide maximum availability, the cluster can automatically adjust the quorum mode as nodes are added or removed from the cluster. The four quorum modes are:

1. **Node Majority**. Each node that is available and in communication can vote. The cluster functions only with a majority of the votes, that is, more than half. This is similar to Majority Node Set (MNS) in Windows Server 2003. This selection would be preferred with an odd number of nodes and if no shared disk/storage is provided.
2. **Node and Disk Majority**. Each node plus a designated disk in the cluster storage (the “disk witness”) can vote whenever they are available and in communication. The cluster functions only with a majority of the votes, that is, more than half. This is the recommended configuration with an even number of nodes and for when shared storage is available.
3. **Node and File Share Majority**. Each node plus a designated file share created by the administrator (the “file share witness”) can vote whenever they are available and in communication. The cluster functions only with a majority of the votes, that is, more than half. This would commonly be implemented in a geographically dispersed failover cluster environment.
4. **No Majority: Disk Only**. The cluster has quorum if one node is available and in communication with a specific disk in the cluster storage. Only the nodes that are also in communication with that disk can join the cluster. This is equivalent to the quorum disk in Windows Server 2003. The disk is a single point of failure, so only select scenarios should implement this quorum mode.

When you install a Windows Server 2008 cluster, a recommended default quorum mode is selected during the installation. For example, if the cluster has an even number of nodes, the **Node and Disk Majority** quorum mode selected by default; if the cluster has an odd number of nodes, the **Node Majority** quorum mode will be selected. The user is not prompted to select a specific type of quorum mode during the installation. Furthermore, when nodes are added and removed from the cluster, the quorum mode can change automatically. This may surprise users who are accustomed to clusters in Windows Server 2003.

When you set up a cluster, it is important to understand the implications of each quorum model and the number of failures (node and/or disk) that a quorum mode can tolerate. You must also evaluate the requirements of the application to select the right quorum mode.

To maintain the uptime of a system, it is important to align the operational practices with the technology that is used to achieve high availability. For example, if your requirement is to restart two nodes in a three-node cluster, and still be available during that time, it may make sense for you to use the **No Majority: Disk Only** quorum mode. However, make sure that the disk quorum uses highly reliable storage (for example, a mirrored disk on a fully redundant storage area network (SAN)).

For further information about the quorum modes and considerations for which mode to choose, see “[Failover Cluster Step-by-Step Guide: Configuring the Quorum in a Failover Cluster](http://technet.microsoft.com/en-us/library/cc770620.aspx)” (http://technet.microsoft.com/en-us/library/cc770620.aspx). For additional discussion and scenario models, see “[Appendix B: Additional Information About Quorum Modes](http://technet.microsoft.com/en-us/library/cc770830.aspx)” (http://technet.microsoft.com/en-us/library/cc770830.aspx).

For further information about Windows Server 2008 failover clustering, see “[Failover Clusters](http://technet.microsoft.com/en-us/library/cc732488.aspx)” (http://technet.microsoft.com/en-us/library/cc732488.aspx) on the Microsoft TechNet Web site.

# Topology and Architecture of a SQL Server Cluster

A SQL Server failover cluster instance that is installed on a Windows Server failover cluster is similar to a stand-alone SQL Server instance that does not use Windows Server failover clustering in terms of the way it runs and appears externally to clients. The key difference is that SQL Server failover clustering leverages the Windows Server failover clustering platform to monitor and provide high availability of the SQL Server instance across redundant nodes. This section describes the SQL Server failover cluster resources and resource groups, in addition to common configurations.

## Failover Cluster Resource Group

A SQL Server failover cluster instance has two parts:

* The local binaries and components that are installed on each possible owner node.
* The shared resources in the resource group that fail over between each possible owner node.

The local binaries and components on each node are similar to those for a stand-alone SQL Server instance. However, at startup, instead of the service being started either automatically or manually by the user, the Windows Server failover cluster service monitors and manages the instance.

Each SQL Server failover cluster instance has a resource group that contains the following set of resources:

* Network Name.
* IP Address.
* One or more shared disks.
* SQL Server Database Engine service.
* SQL Server Agent service.
* SQL Server Analysis Services service, if installed.
* One file share resource, if the FILESTREAM feature is installed.

The network name and IP address resources provide a single identifier that clients can use to connect to the SQL Server instance regardless of which node is currently serving the SQL Server failover cluster instance. When the resource group fails over, these resources register the network name and IP addresses to redirect to the new node that is serving the SQL Server failover cluster instance. This process can be transparent to clients who do not need to change the name or IP address that they are using to connect to SQL Server.

A SQL Server shared disk resource contains all of the system and user data including databases, logs, FILESTREAM files, and integrated full-text search files for the SQL Server failover cluster instance. When a failover occurs, the disks are mounted on a new node. When the SQL Server instance is started on the new node, it goes through recovery as part of the SQL Server startup and maintains access to the same database files that existed when it was running on the previous node.

The shared disks are the single point of failure for a failover cluster instance. Windows Server and SQL Server failover clustering provides redundancy for machines, operating systems, and SQL Server binaries, but requires reliable storage to assure availability of the shared storage. The shared disk resource type can either be the built-in shared disk resource or a custom shared disk resource provided by your storage vendor for more advanced storage configurations such as Storage Area Network (SAN) replication technologies.

The SQL Server, SQL Server Agent, and Analysis Services resources are responsible for bringing their respective services online and offline on each node. The Windows Server failover cluster directs this process in response to health detection messages that relate to each resource.

The SQL Server and SQL Server Agent resources are part of any SQL Server failover cluster’s resource group where the Database Engine feature is installed. The SQL Server Analysis Services resource is a part of any SQL Server failover cluster’s resource group where Analysis Services is installed. Although it is possible to install both Database Engine and Analysis Services in the same resource group, it is generally recommended to install them as separate instances. Analysis Services is not dependent on SQL Server, and it should be installed to a separate resource group for maximum availability and performance.

## Health Detection

A SQL Server failover cluster has two levels of health detection:

* The Windows failover cluster detects that the nodes are online and able to communicate
* Each resource also provides its own specific health detection

SQL Server and Analysis Services rely on the built-in Windows Server failover clustering resources for network name and IP addresses which perform health checks to ensure the network name and IP address are online and properly registered. If the network name or IP address resource is offline, the SQL Server or Analysis Services resources that depend on them will also be in an offline state.

Shared disk health checks are built in to either the Windows failover cluster resource checks or the resource checks provided by a third-party cluster-aware storage vendor. In both cases, these resources will verify that the disk is attached and available.

For the SQL Server resource, the **LooksAlive** check (also referred to as **Basic resource health check** in Windows Server 2008) will verify that the SQL Server service is running on the online node every 5 seconds by default. If the **LooksAlive** check fails, the Windows Server cluster service performs an **IsAlive** check to confirm the failure.

The **IsAlive** check (also referred to as **Thorough resource health check** in Windows Server 2008) runs every 60 seconds and verifies the cached result of an internal **IsAlive** process in the SQL Server resource DLL.

An internal process in the SQL Server resource DLL handles all connection and execution of the underlying **IsAlive** logic. It contains extensive retry logic to verify any connectivity or execution failures. Upon successful connection to the SQL Server, the internal **IsAlive** check executes SELECT @@SERVERNAME every 60 seconds to determine whether SQL Server is online. If this query fails, the internal process runs additional retry logic to avoid stress-related failures. If the retry logic also fails, the internal process shuts down the SQL Server service and updates a cached result that causes the next **LooksAlive** and **IsAlive** checks to fail.

The SQL Server resource DLL is loaded by the Windows cluster service when the SQL Server resource is started. Therefore, the calls that the resource DLL makes are made in the context of the Cluster service. Consequently, the Cluster service must have access to the SQL Server’s **master** database. It is unnecessary to add the Cluster service account or the BUILTIN\Administrators group to the **sysadmin** role. However, the Cluster service account must be a user in the master database which has permissions to execute SELECT @@SERVERNAME. By default, the **public** role has this permission and the **guest** role is enabled for **master,** so simply adding the Cluster service account as a login with no additional permissions is usually sufficient. The SQL Server 2008 installation automatically adds the Cluster service startup account (or the service SID for the Cluster service) as a login during setup.

The Analysis Services service is installed as a generic service and its state (**LooksAlive**/ **IsAlive**) is determined by using standard logic for generic clustered applications.

## Resource and Resource Group Properties

The SQL Server resource group and each resource within the resource group expose their failover properties through the Windows Server Failover Cluster Management Microsoft Management Console (MMC) snap-in. In addition, they expose their failover properties through cluster application programming interfaces (APIs) and the cluster.exe command-line tool.

For more information about available cluster APIs, see “[Failover Cluster APIs (Windows)](http://msdn.microsoft.com/en-us/library/cc296100(VS.85).aspx)” (http://msdn.microsoft.com/en-us/library/cc296100(VS.85).aspx). For more information about cluster.exe commands, see “[Managing a Server Cluster from the Command Line: Server Clusters (MSCS)](http://technet.microsoft.com/en-us/library/cc779044.aspx)” (http://technet.microsoft.com/en-us/library/cc779044.aspx).

**Note:** In the Failover Cluster Management tool in Windows Server 2008, the resource group is called “Services and Applications.” These terms are used interchangeably throughout this document.

### Resource Properties

Each resource may expose several types of properties through Windows Server Failover Cluster Management. For Windows Server 2008, these include:

* **General**. The basic properties for this resource type.
* **Dependencies**. These define what other resources in the group must be online before a particular resource can be brought online.
* **Policies**. These define how the resource responds to a failure and what impact a failure of this resource has on the entire resource group.
* **Advanced policies**. These policies control the possible owners and the health-check intervals.
* **Permissions**. These define share and file permissions for a file-share resource. For example, SQL Server Setup will install a file-share cluster resource to support the FILESTREAM feature, if it is enabled.
* **Registry replication**. These are the registry keys that are replicated across all nodes via the checkpoint process.
* **Properties**. These are custom properties that are defined for the resource type.

For Windows Server 2003, a subset of the list above is included.

Each resource has a set of possible owners, which are the nodes that the resource is allowed to run on. In most cases, you should not change the possible owners manually. However, one exception to this is during the execution of a rolling update where ownership should be changed (see the “Rolling Updates” section below for more information). Otherwise, SQL Server Setup will handle adding and removing possible owners itself during installation, node addition, node removal, and major version upgrades.

Resource dependencies enable you to customize what the resource depends on. Windows Server failover clustering ensures the correct dependency order of startup and shutdown, and terminates any resource that is dependent on a resource that fails or is offline. SQL Server and SQL Server Analysis Services are dependent on the shared disk resources and the network name resource. The network name resource is dependent on the IP address resource, and SQL Server Agent is dependent on the SQL Server Database Engine resource.

Windows Server 2008 adds the ability to specify **OR** dependencies in addition to **AND** dependencies. SQL Server 2008 and earlier versions do not support **OR** dependencies between IP addresses, which is necessary to enable nodes to run on different subnets.

Each resource also enables you to configure failover policies that determine:

* Possible owners. All resources within the same group should have the same possible owners.
* Whether the resource should attempt to restart if it fails.
* The maximum number of restarts to attempt on a given node within the specified period.
* Whether failure of this resource should cause the entire resource group to fail over to another node.
* Pending time-out (the time-out value to let the resource change states).
* The **LooksAlive** check (a basic resource health-check interval).
* The **IsAlive** check (a thorough resource health-check interval).   
  **Note:** Changing this interval does not affect the standard detection, which runs every 60 seconds. It does, however, change the interval at which the Windows Server Failover Cluster service checks the cached result of the SQL Server resource DLL. It is not recommended to change the interval from the default selection.

SQL Server 2008 uses the default policy of attempting to restart once on the same node on the first failure within a 15-minute interval. If a second failure occurs within the 15-minute interval, the entire resource group fails over. Each resource is also configured to affect the group if a restart on the node is unsuccessful. Each resource in the SQL Server resource group should be evaluated to ensure that its failover policy aligns with business requirements. Some customers may decide to disable the “**Affect the group**” setting (also referred to as “**if restart is unsuccessful, fail over all resources in this service or application**” in Windows Server 2008) for resources such as the ones listed below, if they do not want a failure of these resources to cause a failover, resulting in downtime, for SQL Server.

* The SQL Server Agent resource.
* The FILESTREAM file share.
* Microsoft Distributed Transaction Coordinator (MSDTC), if it is installed to the SQL Server resource group.
* SQL Server Analysis Services if it is installed in the same failover cluster instance as the Database Engine.

The pending time-out generally remains at the default value of three minutes; it is not usually necessary to change this value.

All resources have a default **LooksAlive** interval of five seconds. You can alter this value if you want to check whether the service looks alive more frequently. It is recommended to leave the default setting for this value.

SQL Server uses an internal process in the custom resource DLL to manage **IsAlive** logic for SQL Server failover clusters. The internal **IsAlive** logic maintains cached values to indicate the current state of the SQL Server service. The internal timings for this logic cannot be changed. Changing the resource properties for the SQL Server resource within the Cluster Administrator changes the frequency that the Cluster service checks the cached results of the SQL Server resource DLL.

The SQL Server resource exposes custom SQL Server–specific properties for troubleshooting. In Windows Server 2008, these are visible on the **Properties** tab. In Windows Server 2003, you can also view these through the cluster.exe command:

cluster.exe RESOURCE [SQL Engine Resource Name] /PRIVPROPERTIES

For additional information about these properties see the “Properties for the SQL Server Resource” section in the “Troubleshooting SQL Server Failover Clusters” section of this paper.

### Resource Group Properties

Each resource group exposes the following configurable properties:

* Preferred owners order.
* **AntiAffinityClassNames** (this property is not exposed via the Cluster Administrator graphical user interface (GUI)).
* Automatic failback policies.
* Maximum number of retries or failures to attempt within a specified period.

You can use the **preferred owners** setting to specify nodes that are more desirable to run the application on. If you select multiple preferred owners, you can also configure the order of preference. On clusters that have more than two nodes, you should set your preferred owners list to provide a more balanced failover across the cluster.

The **AntiAffinityClassNames** setting enables the user to specify applications that they do not want to run on the same node. The property consists of zero or more arbitrary user-defined strings. Groups that have string names in common are said to be “anti-affined” from each other, which means that they should not be hosted on the same node if possible. For further information about how **AntiAffinityClassNames** works in conjunction with preferred owners, see “[AntiAffinityClassNames(Windows)](http://msdn.microsoft.com/en-us/library/aa369651.aspx)” (http://msdn.microsoft.com/en-us/library/aa369651.aspx).

**Note:** By default, SQL Server 2008 is configured with no preferred owners or values for **AntiAffinityClassNames**. These are optional property settings.

The resource groups also enable you to configure a policy to fail back to a specific “most preferred” node. You can configure the policy to attempt failback immediately after the node is available again, or within a maintenance range period, which is specified in hours. By default, automatic failback is not configured for SQL Server 2008. When you configure failback, using immediate failback can result in SQL Server failover that has less granular control than when allowing an administrator to schedule when to move the group.

You can also specify the maximum number of failover attempts or restarts for each resource group within a specified time before leaving the resource group in a failed state. By default, the SQL Server 2008 resource group is configured to tolerate two failures every six hours.

## Common SQL Server Failover Deployments

There are various architectural deployment scenarios for SQL Server failover clustering. This section describes the most common failover cluster configurations:

* Single failover cluster instance two-node cluster.
* Multi-instance cluster:
  + All nodes with active instances.
  + *n*+1 (*n* instances with one spare node).
* Multi-site failover cluster instance.
* Guest failover cluster.

**Note:** The terms “Active/Passive,” “Active/Active,” or some other combination of “Active” and “Passive” (such as “Active/Active/Passive”), are often used in conjunction with SQL Server failover clustering. This was an accurate definition when you could only install a single SQL Server instance on each node in SQL Server (on versions that predated SQL Server 2000). Therefore, from the application (SQL Server) perspective, any node could be “active” or “passive” in terms of running SQL Server. Today, however, with multiple failover cluster instances able to run on a single node, and the potential of many more nodes within the cluster, these terms can be imprecise and ambiguous when used to describe the configuration of the SQL Server instances and the nodes they are running on. It is more accurate to describe each failover cluster instance separately in terms of where it is currently running and what its possible owners are. Furthermore, the “Active/Active” term can give the misleading impression that some load balancing is occurring across nodes or across SQL Server instances.

### Single Failover Cluster Instance

The most common failover cluster configuration is a two-node cluster with a single SQL Server failover cluster instance. This arrangement provides a dedicated machine to protect against the failure of a single node. It is strongly recommended that both machines in this configuration have the same hardware provisioned to ensure that your workload continues to run after a failover without undesirable performance constraints.

If there is a need to tolerate additional machine failures you can increase the number of nodes within the cluster. SQL Server 2008 Enterprise Edition will be required if you want the SQL Server failover cluster to be able to fail over to more than two nodes. You should also consider the quorum type of the underlying Windows Server cluster to ensure that it can tolerate the desired number of failures.

Sometimes provisioning dedicated spare machines for each SQL Server failover cluster instance is too costly and you may be willing to trade off availability for hardware utilization by combining multiple SQL Server failover cluster instances on a single failover cluster. This is referred to as a multi-instance failover cluster.

### Multi-Instance Failover Cluster

There are two common types of multi-instance clusters. The first is where all nodes are running one or more failover cluster instances. When a node fails, all of the failover cluster instances that it is hosting will fail over to another node. The primary consideration for this configuration is whether each node has enough hardware capacity to run the peak workloads for the failover cluster instances that it already hosts, plus the failover cluster instances that it will take over in the event of the failure of another node. This issue can be compounded if you need to tolerate multiple node failures. You can use the preferred owners and **AntiAffinityClassNames** resource group properties for greater control of the balancing of SQL Server instances during a failure scenario. As in any availability solution, you should ensure that there is enough spare hardware capacity to continue serving the application workloads in the event of machine failure(s). Alternatively, you should be prepared to run under a modified service level agreement (SLA) for the duration that the node is unavailable. See *Considerations for SQL Server Consolidation with a Failover Cluster* in “[Installing a SQL Server 2008 Failover Cluster](http://msdn.microsoft.com/en-us/library/ms179410.aspx?)” (http://msdn.microsoft.com/en-us/library/ms179410.aspx?) for example scenarios of what to consider in a multi-instance failover cluster.

The second common multi-instance cluster configuration is what is often referred to as an “*n*+1” configuration. In a number of nodes (*n*), each node is running one or more active failover cluster instances, and a dedicated spare node (+1) is available to take on the workload of any other node upon failure. This configuration reduces potential capacity considerations during failure that arise when all nodes are active. It also generates greater machine utilization because you can use the dedicated spare machine across multiple failover cluster instances.

You can, of course, increase the number of spare nodes to any number, as long as you stay within the SQL Server supported limits of nodes for your operating system version. Using preferred owners and **AntiAffinityClassNames** on the resource group gives greater control over SQL Server failover cluster instance allocation or balancing during multiple failures.

### Multi-Site Failover Cluster

A multi-site failover cluster includes nodes that are dispersed across two physical sites or data centers, with the aim of providing a disaster recovery solution to protect against site failure. Sometimes multi-site failover clusters are called *geographically dispersed failover clusters* or *stretch clusters*. In addition to Windows Server failover clustering, SQL Server relies on two key pieces of technology in a multi-site failover cluster:

* Storage replication technology (for example, SAN replication) to ensure that the dependent disks can fail over.
* Stretch virtual LAN (VLAN) to provide a single subnet across all nodes.

When you choose a storage replication technology for multi-site failover clustering, ensure that it meets SQL Server storage requirements and that your databases and their logs are synchronously and consistently replicated in order to avoid data loss. If your database and log files reside on different volumes, you should work with your storage solution to ensure consistency between the data and log file when SQL Server recovers the database after a failover. The key concern is that SQL Server write ordering requirements are adhered to and that the data is consistent even when network interruptions or failures occur.

Although features of Windows Server 2008 added functionality to support multi-subnet failover clusters, SQL Server 2008 does not support this additional functionality and requires that all nodes reside on the same subnet. Different sites or data centers often do not share a single subnet, so implementing a multi-site failover cluster for SQL Server 2008 generally requires the use of a stretch VLAN solution to expose a single subnet on all nodes of a SQL Server failover cluster instance. This is an appropriate configuration (single subnet) for a multi-site SQL Server failover cluster.

By using the storage replication and stretch VLAN technologies to provide common network and storage access to all nodes, a multi-site failover cluster will appear to SQL Server 2008 to be similar to a standard single-site failover cluster.

When you configure a multi-site failover cluster, you must also consider:

* The effect that site failures or a loss of communication can have on quorum.   
  Configuring quorum settings on a multi-site failover cluster is the same as configuring them for a single-site failover cluster. However, a multi-site cluster configuration is designed to protect against a site failure and is also susceptible to a loss of communication between physical sites. Therefore, typically, the quorum is configured so that either site can run should the other fail.
* The configuration of heartbeat settings to tolerate increased latency.   
  Depending on the latency between the sites in your multi-site failover cluster, you may want to increase the tolerance for Windows Server failover clustering inter-node heartbeat to avoid latency falsely being detected as stress.

See “[Requirements and Recommendations for a Multi-Site Failover Cluster](http://technet.microsoft.com/en-us/library/dd197575.aspx)”(http://technet.microsoft.com/en-us/library/dd197575.aspx) for additional resources about configuring multi-site clustering, including specific details about how to configure the heartbeat.

You can, of course, combine multi-site failover cluster instances into the multi-instance failover cluster.

**Note:** Some multi-site failover cluster implementations have experienced failures of SQL Server Setup during disk qualification. This is due to how SQL Server 2008 Setup investigates disk resources and the resources that are dependent on those disks. See the Developer Note below for further technical information about this. If you believe you are experiencing this problem, consider:

1. Using the Failover Cluster Management MMC to verify that all disks are part of the same group and that there are no nondisk resource dependencies to any of the disks.
2. Contact the hardware partner for advice on the appropriate course of action.

**Developer Note:** SQL Server 2008 Setup does not look at the physical storage resource type. Instead, the CLUS\_RESCLASS\_STORAGE flag is investigated when it looks for storage resources and checks dependencies. If you are getting blocked during disk qualification, it is likely that a resource in the dependency chain does not have that flag set. If the resources on both sides of the dependencies are CLUS\_RESCLASS\_STORAGE, then they will not be blocked by SQL Server Setup. Please note that this applies to the entire storage dependency graph. SQL Server Setup walks the entire graph to make sure that it has only CLUS\_RESCLASS\_STORAGE. Even if the dependencies that are directly connected to the disk resource are all valid, the whole chain is disqualified if the replication solution is referenced by, for example, a service. Any graph that contains just CLUS\_RESCLASS\_STORAGE should pass the check.

### Guest Failover Cluster

A *guest failover cluster* is a SQL Server failover cluster inside a virtual machine where the nodes are running as virtual machines. Both SQL Server 2005 and SQL Server 2008 support guest failover clustering, provided that it meets the following requirements:

1. The host operating system is running in one of the following virtualization environments:

* Windows Server 2008 with Hyper-V™.
* Microsoft Hyper-V™ Server 2008.
* Configurations that are certified through the Server Virtualization Validation Program (SVVP).

1. The operating system that is running in the virtual machine (the “guest operating system”) is Windows Server 2008 or higher.
2. The virtualization environment meets the requirements of Windows Server 2008 failover clustering as documented in “[The Microsoft Support Policy for Windows Server 2008 Failover Clusters](http://support.microsoft.com/kb/943984)” (http://support.mirosoft.com/kb/943984).

Each of the SQL Server guest cluster nodes can run on the same physical host machine or different physical host machines. If SQL Server guest clustering is running on the same host, its availability will be affected when the host becomes unavailable. Thus for maximum availability and protection, consider running the active node and standby node of a SQL Server guest cluster on different physical host machines.

For support policy and details, please refer to “[Support policy for Microsoft SQL Server products that are running in a hardware virtualization environment](http://support.microsoft.com/kb/956893)” (http://support.microsoft.com/kb/956893).

## Multiple SQL Server Versions Within the Same Windows Server Cluster (SQL Server 2000 and 2005 with SQL Server 2008)

As mentioned above, you can install multiple SQL Server failover cluster instances on the same Windows Server failover cluster in what is often called a *multi-instance cluster*. In this configuration, each instance has its own set of binaries, which can be at different service pack (SP), cumulative update, or hotfix versions, with the exception of some shared features that are covered later in this topic.

On a Windows Server cluster, coexistence of SQL Server 2008 with SQL Server 2000 SP4 or later, and SQL Server 2005 SP2 (being the lowest currently supported service pack for SQL Server 2005) or later versions is supported. Of course, it is recommended to run each version with the most recent service pack available to ensure that your application is protected by the most up-to-date fixes available. See the [Microsoft Support Lifecycle](http://support.microsoft.com/lifecycle/) (http://support.microsoft.com/lifecycle/) page for currently supported versions including minimum supported service pack levels for each version.

There is one caveat: if you need SQL Server 2000 along with SQL Server 2005 or SQL Server 2008, you must install SQL Server 2000 SP4 on each failover cluster node before you install SQL Server 2005 or SQL Server 2008. This is due to the version of the SQL Server failover cluster resource DLL and its version compatibility. This limitation also means that you cannot add nodes to, or remove them from, a SQL Server 2000 failover cluster instance after installing SQL Server 2005 or SQL Server 2008. If possible, it is recommended, but not required, that you do one of the following:

1. Install SQL Server 2000 failover cluster instances on nodes without SQL Server 2005 or SQL Server 2008.
2. Upgrade all SQL Server 2000 failover cluster instances to SQL Server 2005 or SQL Server 2008.

You can install SQL Server 2005 and SQL Server 2008 failover cluster instances in any order that you want. When you install SQL Server 2008 through the user interface (UI), the **Feature Selection** screen is separated into two main sections: **Instance Features** and **Shared Features**.

**Instance Features** refers to the components that are installed once for each instance so that you have multiple copies of them (one for each instance) on each node that is a possible owner of an instance. Each instance can be a separate version that has a different patch level. When you install a patch, whether it is a service pack, a hotfix, or a cumulative update, it updates only the files for one instance on one node, plus the shared features if they have not already been updated.

**Shared Features** refers to features that are common across all instances on a given machine. Shared features include the client tools, the Browser service, and the failover cluster resource DLL. Each of these shared features is designed to be backward compatible with supported SQL Server versions (service packs, cumulative updates, and hotfixes) that can install side by side. There is, at most, one copy of each shared feature for each version on each node.

## Advantages of SQL Server 2008 on Windows Server 2008 Clustering

Although a Windows Server 2003 failover cluster supports the installation of SQL Server 2008 failover clustering, Windows Server 2008 failover clustering introduces several important new features that SQL Server 2008 failover clustering takes advantage of:

* **The removal of the validation requirement for cluster solution hardware and components with the Hardware Compatibility List (HCL) and Windows Server Catalog**

Before Windows Server 2008, Microsoft server failover clusters were only deemed “supported” if the entire cluster solution was listed in the Windows Server Catalog under the **Cluster Solutions** category. Matching a given cluster hardware solution against the Windows Server Catalog could be challenging, and often required further communication and validation with the solution vendor to determine supportability.

Windows Server 2008 removes this requirement. Instead of checking with the Windows Server Catalog, your Windows Server 2008 cluster solution must instead pass validation by using the Windows Server 2008 Cluster Validation Tool. Before cluster configuration, this tool scans the server nodes and storage that you intend to use for your cluster solution. The tool validates hardware and reports on any blocking issues that may impact the support of the failover cluster, detailing issues across storage, the operating system, hardware, and network components. If all hardware components are certified for Windows Server 2008 and the validation passes all tests, the failover cluster solution is supported from Microsoft’s point of view.

* **Internet small computer system interface (iSCSI) support**

Expanding SQL Server 2008 failover cluster storage options beyond Fiber Channel and Serial Attached SCSI (SAS) storage connections, Windows Server 2008 failover clusters support iSCSI storage connections.

* **IP version 6 (IPv6) support**

Windows Server 2008 DNS servers support IPv6, which expands IP address size to 128 bits (rather than 32 bits with IPv4). SQL Server 2008 failover clustering natively supports IPv6.

* **Dynamic Host Configuration Protocol (DHCP) support**

DHCP is now supported for Windows Server 2008 failover cluster server nodes. The cluster service is responsible for managing the IP addresses, and SQL Server 2008 failover clustering supports this scenario. However, from a system stability perspective, it is still recommended to use static IP addresses for the Windows Server 2008 failover cluster server nodes rather than DHCP when hosting SQL Server 2008 failover clusters. Any service or application dependencies on an IP address that fails to renew could cause a failure in the IP address resource, in addition to a failure with the clustered service or application.

* **Service SIDs**

SQL Server 2005 failover clustering introduced the requirement of designating domain groups during installation. Domain groups were used in the installation process to provision ACLs and required permissions. These designated domain groups required that the SQL Server service accounts already be members of the domain groups. If this was not the case, permissions were needed to add them as members. This requirement added administrative difficulty to the overall installation process because it became hard to create a common separation of duty between database administrator (DBA) teams and Active Directory® administrators.

When you install a SQL Server 2008 failover cluster on a Windows Server 2008 failover cluster, domain groups are no longer required. Instead, you can choose to designate the use of service SIDs. When you install a SQL Server 2008 failover cluster, you designate “Use service SIDS” to bypass the need to provision domain groups. Service SIDs, introduced in Windows Server 2008 and Windows Vista, enables you to bind permissions directly to a Windows service.

* **Supported number of cluster nodes**

The 64-bit edition of Windows Server 2008 Enterprise expands the number of nodes that are supported in a single cluster to 16. On Windows Server 2003, only eight nodes were supported.

Although many of the new features that were introduced in Windows Server 2008 failover clustering are available with SQL Server 2008, a few Windows Server 2008 features are not currently supported or exposed to SQL Server 2008:

* **Installation on Windows Server 2008 Server Core**

SQL Server 2008 failover clusters and stand-alone instances of SQL Server 2008 are *not* supported on Windows Server 2008 Server Core.

* **Separate subnets for cluster nodes**

Although Windows Server 2008 failover clusters support separate subnets for cluster nodes, SQL Server 2008 failover clusters do not. Due to the popular demand for this feature, it is expected that multi-subnet support will be added in a future version of SQL Server.

# SQL Server Installation and Setup

The installation process for SQL Server 2008 failover clusters has changed significantly. The focused and discrete installation steps within the setup of the failover cluster help to minimize failures. They also ensure that you can accomplish more discrete tasks and installation options during SQL Server 2008 failover cluster setup. Unlike earlier versions, when you install or upgrade a SQL Server 2008 failover cluster, you must run the setup process for SQL Server individually on each node of the failover cluster. To add a node to an existing SQL Server failover cluster, you must run SQL Server Setup on the node that is to be added to the SQL Server failover cluster instance. Unlike SQL Server 2005, the node that needs to be managed or modified is where the SQL Server Setup should be executed from.

Before you begin the SQL Server installation, validate the hardware configuration and the functionality of the underlying Windows Server failover cluster by confirming that all resources can fail over and come online. As discussed in the previous section, you can utilize the cluster validation wizard for Windows Server 2008 and the [Windows Server Catalog](http://www.windowsservercatalog.com/results.aspx?&chtext=&cstext=&csttext=&chbtext=&bCatID=1291&cpID=0&avc=10&ava=0&avq=0&OR=1&PGS=25&ready=0) to search for complete cluster solutions for previous Windows Server versions. Also, before the installation, you should reference the following topics in SQL Server 2008 Books Online:

* [Planning a SQL Server Installation](http://msdn.microsoft.com/en-us/library/bb500442.aspx) (http://msdn.microsoft.com/en-us/library/bb500442.aspx)
* [Before Installing Failover Clustering (see especially the “Preinstallation checklist” section)](http://msdn.microsoft.com/en-us/library/ms189910.aspx) (http://msdn.microsoft.com/en-us/library/ms189910.aspx)

**Important:** If you are applying Service Pack 1 for SQL Server 2008 or later, consider using the new slipstream functionality, not only to reduce the installation time but also to have the benefit of setup fixes. If you are not going to install a service pack, you should apply cumulative updates for SQL Server Setup before you install SQL Server 2008. For more information about known issues and detailed instructions, see “[How to update or slipstream an installation of SQL Server 2008](http://support.microsoft.com/kb/955392)”( http://support.microsoft.com/kb/955392).

## Installation Options for a SQL Server Failover Cluster

There are two options for installing a SQL Server 2008 failover cluster: Integrated and Advanced/Enterprise installation. Each option is designed to handle different scenarios, although each option is capable of delivering exactly the same end result. Both installation options are available in the SQL Server 2008 Setup Wizard and via command-prompt installation.

### Integrated Installation

Integrated installation satisfies the most common requirements for SQL Server deployments. Integrated installation is used for creating a SQL failover instance, adding a node to an existing cluster, and removing a node from an existing cluster regardless of the original installation option.

SQL Server integrated failover cluster installation consists of the following steps:

1. Create and configure a single-node SQL Server failover cluster instance. When configuration of the node is successful, you have a fully functional failover cluster instance. At this point, it does not have high availability because there is only one node in the failover cluster.
2. On each node to be added to the SQL Server failover cluster, run Setup with Add Node functionality to add that node.

For step-by step instructions for installing a SQL Server 2008 failover cluster instance by using the integrated installation option, see [Appendix A: How to Create a SQL Server 2008 Failover Cluster - Integrated Installation with Add Node](#_Appendix_A:_Installing).

### Advanced/Enterprise Installation

The Advanced/Enterprise installation option enables users to take advantage of a two-phase SQL Server failover cluster installation process. This is beneficial in enterprise deployment solutions that use technologies such as Systems Management Server (SMS) or scripted deployments (although this is not required). In the first phase, the setup prepares failover cluster instances simultaneously by distributed deployment technology such as SMS or Windows Powershell™, although you can run setup individually on each node, too. Phase two completes the failover cluster creation across all of the prepared instances simultaneously. The Prepare step installs the binaries and files that are necessary for SQL Server to run on each node, and the Complete step joins the prepared nodes to create a SQL Server clustered instance.

SQL Server Advanced/Enterprise failover cluster installation consists of the following steps:

1. **Prepare**. Running the Prepare Failover Cluster setup on one node creates the Configuration.ini file that lists all of the settings that were specified. On the additional nodes to be prepared, instead of following these steps, you can supply the auto-generated Configuration.ini file from the first node as an input to the Setup command line. For more information, see the “Using a Configuration File” section below*.* This step prepares the nodes to be clustered, but there is no operational instance of SQL Server at the end of this step.
2. **Complete**. After the nodes are prepared for clustering, run Setup on the node that currently owns the shared disk resource. This step configures the failover cluster instance and finishes the installation. At the end of this step, you will have an operational SQL Server failover cluster instance and all of the nodes that were prepared previously for that instance will be the possible owners of the newly created SQL Server failover cluster.

**Note:** The Prepare step does not require an underlying Windows Server failover cluster to be established. The Complete step, however, does require that the underlying Windows Server failover cluster exists. If it does not, setup provides an error and exit.

For step-by-step instructions to install a SQL Server 2008 failover cluster instance by using the Advanced/Enterprise installation option, see [Appendix B: Installing a SQL Server 2008 Failover Cluster Using Advanced/Enterprise Installation](#_Appendix_B:_Installing) of this document.

## Adding a Node to a Failover Instance

SQL Server 2008 provides specific installation steps to incorporate a node within the cluster as a potential node for a SQL Server failover instance to run on. To add a cluster node that is not already configured as part of a SQL Server instance, you must run setup on the node to add it to the SQL Server failover instance. This process copies the binaries to the local node and configures the node to be part of the SQL Server cluster. By default, when a node is added to a failover cluster, it is added to the end of the list of the available nodes and becomes available for failover.

For step-by-step instructions to add a node to an existing SQL Server 2008 failover cluster, see [Appendix C: Add Node](#_Appendix_C:_Add) of this document.

## Removing a Node from a Failover Instance

Sometimes it is necessary to remove a node from the list of servers that are configured to host a SQL Server failover cluster instance. Scenarios where this can be helpful include upgrading hardware on particular nodes or as part of the process of addressing a node failure.

Removing a cluster node removes the binaries and the SQL Server configuration information from that node. It also eliminates it as a host of the SQL Server failover instance from which it was removed. The removal process itself does not introduce any downtime for the SQL Server failover instance.

For step-by-step instructions to remove a node from an existing SQL Server 2008 failover cluster, see [Appendix D: Removing a Node](#_Appendix_D:_Removing) of this document.

## Uninstalling a SQL Server Instance

To completely uninstall an instance of SQL Server 2008, each node needs to be removed as part of the SQL Server failover cluster. When SQL Server is removed from the last node of the cluster, the instance and all of the SQL Server resources that are associated with it will be removed from the cluster. Physical disk resources and user data are left intact.

## Unattended Installation of SQL Server 2008 Failover Clustering

Using the attended GUI-based installation interface is suitable for environments that require infrequent installations. However, large enterprise environments that have hundreds or even thousands of SQL Server instances will probably need to streamline the installation and maintenance process by using an unattended installation, which is based on the command prompt. This section describes how to perform unattended installation tasks that are related to failover clusters by using the command prompt. The following section describes how to use a configuration file in conjunction with a command-prompt installation.

Unattended command-prompt installations allow a variety of actions, including:

* Install
* Upgrade
* Remove
* Uninstall
* Repair
* Rebuild Database
* Patch
* Remove Patch
* Edition Upgrade

Command-prompt installations also include a set of actions that are specific to SQL Server 2008 failover clusters, including:

* **InstallFailoverCluster**

This action creates the first SQL Server failover cluster instance that runs on a single cluster node by using the integrated installation setup process.

* **AddNode**

This action adds a cluster node as an available failover partner to a SQL Server failover cluster instance that is already installed as part of the integrated installation setup process.

* **RemoveNode**

This action removes the server node from the list of available failover partners, removing local SQL Server binaries and services. If you perform this action on the last available node of a SQL Server 2008 failover cluster, the SQL Server 2008 failover cluster instance is removed entirely.

* **PrepareFailoverCluster**

This action prepares the node with necessary binaries and services without bringing the SQL Server instance online. It is run as part of the advanced/enterprise installation process. You should execute this option for each server node that should be available to the SQL Server 2008 failover cluster instance before issuing the **CompleteFailoverCluster** action.

* **CompleteFailoverCluster**

This action brings the SQL Server 2008 failover cluster instance online on the active cluster server node that owns the shared disk that will host the SQL Server database files. It is run as the last step of the advanced/enterprise installation process.

The method for initiating an unattended command-prompt installation involves calling the setup.exe executable file from the context of the installation directory, followed by a series of parameter and value switches that use intuitive naming standards.

Parameters are preceded by a slash mark (/) and are followed by an equal sign (=) and the associated parameter value. For example:

/INSTANCENAME="CAESAR"

Executing **setup.exe /?** at a command prompt generates a comprehensive description of available command-prompt installation options. The following table shows an abridged list of options that apply to failover clustering operations that will be demonstrated in this white paper.

|  |  |
| --- | --- |
| **Option** | **Description** |
| AGTSVCACCOUNT | Specifies a domain user name for the SQL Server Agent service. |
| AGTSVCPASSWORD | Specifies the password for the SQL Server Agent service account. This is not required for a system account. |
| CONFIGURATIONFILE | Specifies the configuration file to be used for setup. |
| FAILOVERCLUSTERDISKS | Specifies a cluster shared disk to associate with the SQL Server failover cluster instance. |
| FAILOVERCLUSTERGROUP | Specifies the name of the cluster resource group for the SQL Server failover cluster instance. |
| FAILOVERCLUSTERIPADDRESSES | Specifies an encoded IP address. The encodings are semicolon-delimited (;) and follow the format <IP Type>;<address>;<network name>;<subnet mask>. Supported IP types include DHCP, IPv4, and IPv6. |
| FAILOVERCLUSTERNETWORKNAME | Specifies the name of the SQL Server failover cluster instance. This name is the network name that is used to connect to SQL Server services. |
| FEATURES | Specifies features to install, uninstall, or upgrade. The list of top-level features includes SQL, AS, RS, IS, and Tools. The SQL feature installs the Database Engine, replication, and full text. The Tools feature installs management tools, SQL Server 2008 Books Online, Business Intelligence Development Studio, and other shared components. |
| INSTALLSHAREDDIR | Specifies the root installation directory for native shared components. |
| INSTALLSQLDATADIR | The Database Engine root data directory. |
| INSTANCEDIR | Specifies the instance root directory. |
| INSTANCEID | Specifies the Instance ID for the SQL Server features that you have specified. SQL Server directory structure, registry structure, and service names will reflect the instance ID of the SQL Server instance. |
| INSTANCENAME | Specifies a default or named instance. MSSQLSERVER is the default instance for editions of SQL Server other than Express; for Express, the default instance is SQLExpress. This parameter is required when you install the SQL Server Database Engine (SQL), Analysis Services (AS), or Reporting Services (RS). |
| SAPWD | Specifies a password for the SQL Server ‘sa’ account. |
| SECURITYMODE | Specifies the authentication mode. The default is Windows Authentication. Use "SQL" for Mixed Mode Authentication. |
| SQLBACKUPDIR | Specifies the default directory for the Database Engine backup files. |
| SQLCOLLATION | Specifies a Windows collation or a SQL collation to use for the Database Engine. |
| SQLSVCACCOUNT | Specifies a domain account for the SQL Server service. |
| SQLSVCPASSWORD | Specifies a SQL Server service password. Required only for a domain account. |
| SQLSYSADMINACCOUNTS | Specifies a Windows account(s) to provision as a SQL Server system administrator. |

When you perform a command-prompt installation for a failover cluster, you need to have local administrator rights on the server nodes, and the permission to log in as a service and act as part of the operating system. The following example uses the integrated installation method to create the first node of a SQL Server cluster instance and the SQL Server 2008 failover cluster instance itself. The specific action that is selected is *InstallFailoverCluster*.

setup.exe /q /ACTION=InstallFailoverCluster

/FEATURES=SQL

/INSTANCENAME="CAESAR"

/INSTANCEID="CAESAR"

/INSTANCEDIR="C:\Program Files\Microsoft SQL Server" /INSTALLSHAREDDIR="C:\Program Files\Microsoft SQL Server" /SQLSVCACCOUNT="ROME01\DMNSQLSRV01”

/SQLSVCPASSWORD="XYZ12345"

/AGTSVCACCOUNT="ROME01\ DMNSQLSRVAGT01"

/AGTSVCPASSWORD="XYZ12345"

/INSTALLSQLDATADIR= "S:\MSSQLSERVER" /SQLCOLLATION="SQL\_Latin1\_General\_CP1\_CS\_AS" /FAILOVERCLUSTERGROUP="CAESAR"

/FAILOVERCLUSTERDISKS="Cluster Disk 3" /FAILOVERCLUSTERIPADDRESSES="IPv4;172.29.10.160;Cluster Network 1;255.255.248.0"

/FAILOVERCLUSTERNETWORKNAME="SQL" /SQLSYSADMINACCOUNTS="ROME01\Administrator"

/SECURITYMODE=SQL

/SAPWD="Yukon900"

/INSTALLSHAREDWOWDIR="C:\Program Files (x86)\Microsoft SQL Server"

Note that the previous example was illustrative, and requires modification for your specific environment. Also note that the following syntax will not work with slipstream installations, which require a PCUSOURCE parameter. See “[How to update or slipstream an installation of SQL Server 2008](http://support.microsoft.com/kb/955392)” for a description of the slipstream process.

The “/q” switch in the previous example directs setup to operate in an unattended mode (no user interface). After the single-node SQL Server 2008 failover cluster instance is installed, you can then add additional nodes by using the ACTION type of *AddNode*.

setup.exe /q /ACTION=AddNode /INSTANCENAME="CAESAR" /SQLSVCACCOUNT="ROME01\DMNSQLSRV01" /SQLSVCPASSWORD="XYZ12345" /AGTSVCACCOUNT="ROME0 1\DMNSQLSRVAGT01" /AGTSVCPASSWORD="XYZ12345"

Notice that, when adding a node in the previous example, it required far fewer parameters than when having to add a new SQL Server 2008 failover cluster instance by using the *InstallFailoverCluster* action. You should execute the *AddNode* action for each server cluster node that you want to make available for the SQL Server failover cluster to fail over to. For example, if you have a four-node failover cluster, after executing an *InstallFailoverCluster* action, you would execute an *AddNode* action unattended installation for each of the other three nodes.

In a similar way to adding a node, removing a node requires very few parameters.

setup.exe /q /ACTION=RemoveNode /INSTANCENAME="CAESAR"

To completely uninstall a SQL Server 2008 failover cluster instance, you execute the *RemoveNode* action for each node where the SQL Server 2008 failover cluster instance was added. For example, for a four-node cluster, you would execute *RemoveNode* four times, one for each node. After all of the nodes are removed, the SQL Server 2008 failover cluster instance is effectively uninstalled.

## Using a Configuration File

Instead of designating all setup parameters in setup from the command line, you can use a configuration file that contains your desired setup options. A configuration file is automatically created whenever you use the setup wizard. It is named ConfigurationFile.ini and is created in the <drive>:\Program Files\Microsoft SQL Server\100\Setup Bootstrap\Log\<YYYYMMDD\_HHMMSS\ directory. The file is generated whether you complete an actual setup or not. Therefore, you can generate the file without initiating a setup by stepping through the setup dialog boxes and then canceling the setup before completion of the **Ready to Install** dialog box.

When the configuration file is automatically generated, it contains name/value pairs that represent the various installation options and settings that were selected using the user interface. (Passwords are the exception here; you must designate them manually for security reasons.) At the time of writing, when you perform a SQL Server 2008 failover cluster attended installation, the FAILOVERCLUSTERIPADDRESSES setup option is not populated in the Configuration.ini file after the installation is complete. To address this issue, you must manually add the FAILOVERCLUSTERIPADDRESSES setup option to your configuration file.

You can use the generated ConfigurationFile.ini file as an alternative to designating all parameters in the command prompt session for an unattended installation, or to troubleshoot a failed installation.

The following example shows an unattended installation (using integrated installation) of a new SQL Server 2008 failover cluster instance by using a configuration file.

Setup.exe /q /Configurationfile="C:\temp\ConfigurationFile.ini"

The configuration file in this example contains the various parameters and values that are required to create a new single-node instance failover cluster, using integrated installation. The following example shows the abridged configuration file (with comment lines removed).

;SQLSERVER2008 Configuration File

[SQLSERVER2008]

ACTION=InstallFailoverCluster

FEATURES=SQL

INSTANCENAME="CAESAR"

INSTANCEID="CAESAR"

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

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

SQLSVCACCOUNT="ROME01\DMNSQLSRV01"

SQLSVCPASSWORD="abc12345!#$"

AGTSVCACCOUNT="ROME01\DMNSQLSRVAGT01r"

AGTSVCPASSWORD="abc12345!#$"

INSTALLSQLDATADIR="S:\MSSQLSERVER"

SQLCOLLATION="SQL\_Latin1\_General\_CP1\_CS\_AS"

FAILOVERCLUSTERGROUP="CAESAR"

FAILOVERCLUSTERDISKS="Cluster Disk 3"

FAILOVERCLUSTERIPADDRESSES="IPv4; 172.29.8.124;Cluster Network 1;255.255.248.0"

FAILOVERCLUSTERNETWORKNAME="SQL"

SQLSYSADMINACCOUNTS="ROME01\Administrator"

SECURITYMODE=SQL

SAPWD="@123abcde##"

INSTALLSHAREDWOWDIR="C:\Program Files (x86)\Microsoft SQL Server"

By default, passwords are not populated in the configuration file, so you will need to add them manually. Also, notice that the parameter names are identical to the command-prompt setup switches. In this example, after using the configuration file to create the initial instance, you can choose to add nodes to the instance by designating parameters via the command prompt, or you can again choose to use a configuration file.

The following example demonstrates adding a node to an existing failover cluster.

Setup.exe /q /Configurationfile="C:\temp\ConfigurationFile.ini"

Notice that the unattended installation syntax was identical to the previous example, even though we are designating a different action in the configuration file. The *AddNode* action is designated within the body of the configuration file, along with the parameters that are required to add a node for an existing failover cluster, as the following example shows.

;SQLSERVER2008 Configuration File

[SQLSERVER2008]

ACTION=AddNode

INSTANCENAME="CAESAR"

SQLSVCACCOUNT="ROME01\DMNSQLSRV01"

SQLSVCPASSWORD="XYZ12345"

AGTSVCACCOUNT="ROME01\DMNSQLSRVAGT01"

AGTSVCPASSWORD="XYZ12345"

Finally, the following example demonstrates the removal of a cluster node by using a configuration file. Using this method rather than the command prompt does not save much typing, but both options are available for you to choose from.

Setup.exe /q /Configurationfile="C:\temp\ConfigurationFile.ini"

The abridged configuration file that is used to remove a node for a failover cluster is shown in the following example.

;SQLSERVER2008 Configuration File

[SQLSERVER2008]

ACTION="RemoveNode"

INSTANCENAME="CAESAR"

FAILOVERCLUSTERNETWORKNAME="SQL"

By using configuration files, you can modularize the various unattended installation actions you want to perform and keep the parameters out of the command prompt. A configuration file is also generated when you use the Setup UI, so you can reuse the generated configuration file, or simply use it as a reference to understand how your SQL Server failover cluster was set up.

## Post-Installation Considerations

After installation is complete, but before deploying the cluster live in a production environment, there are several issues to consider.

### Adding Additional Cluster Disks for SQL Server

Although SQL Server 2008 failover cluster setup enables you to choose multiple disks for SQL Server data and log files, you may need to add additional cluster disks, which were not configured initially during setup, for use with SQL Server. For further instructions about how to add the dependencies, see “[How to: Add Dependencies to a SQL Server Resource](http://msdn.microsoft.com/en-us/library/ms177447.aspx).”

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

You can use mount points with SQL Server 2008 cluster nodes. For more information, see “[How to configure volume mount points on a Microsoft Cluster Server](http://support.microsoft.com/default.aspx?scid=kb;en-us;Q280297).” (http://support.microsoft.com/default.aspx?scid=kb;en-us;Q280297)

### Verify SQL Server Resource Failover

After installation is complete, you must properly test it to ensure that it works as expected. Follow these steps to do this:

1. Start Cluster Administrator.
2. Right-click the resource group that contains the SQL Server or Analysis Services installation, click **Move Group**, and then click another node in the cluster.   
   SQL Server should now stop on the node that 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 Analysis Services IP address from all nodes within the cluster, and from a computer outside the cluster, to ensure that IP address resolution occurs correctly.
5. Ping the SQL Server and Analysis Services name from all nodes within the cluster, and from a computer outside the cluster, to ensure that there are no problems with name resolution.

## SQL Server Management Tools and Client Installation

If SQL Server client tools, such as SQL Server Management Studio, are installed during setup, they are installed as a shared resource. Therefore, only one set of client tools will exist on any given node of the cluster. These client tools are shared across all instances of SQL Server that are installed on the cluster.

SQL Server client tools are chosen as a specific part of the SQL Server installation. If the tools are installed on the first node of the SQL Server cluster, they are automatically added to any nodes that may be added later to the SQL Server cluster by using Add Node. This behavior differs from SQL Server 2005 Cluster Setup, which only installed the client tools on the local node.

If you do not install the SQL Server 2008 client tools when you initially install the SQL Server, you may install them afterwards by using the options described below.

Via the SQL Server Setup Program

1. Start SQL Server 2008 Setup.
2. On the Installation page, click **New SQL Server stand-alone installation or add features to an existing installation.** (Do not click **New SQL Server failover cluster installation**.)
3. On the Installation Type page, click **Perform a new installation of SQL Server 2008**. (Do not click **Add features to an existing instance of SQL Server 2008**.)
4. On the Feature Selection page, select the tools that you want to install and continue through the rest of the setup process.

### Using the Command-Line Setup

To install the basic SQL Server management tools, run the command in the following example on the node where you want to install the SQL client tools.

Setup.exe /q /ACTION=INSTALL /FEATURES=SSMS

This installs SQL Server Management Tools – Basic, which includes the following:

* SQL Server Management Studio support for the SQL Server Database Engine.
* SQL Server Express.
* The sqlcmd tool.
* The SQL Server Windows PowerShell provider.

To install the complete SQL Server Management Tools, run the command in the following example on the node where you want to install the SQL Server client tools.

Setup.exe /q /ACTION=INSTALL /FEATURES=ADV\_SSMS

This installs SQL Server Management Tools – Complete, which includes the following components in addition to the components in the Basic version:

* + SQL Server Management Studio support for Reporting Services, Analysis Services, and Integration Services.
  + SQL Server Profiler.
  + Database Engine Tuning Advisor.

## Removing the SQL Server Client Tools

Removing the SQL Server client tools is a separate operation from removing a SQL Server instance from a node of a cluster. Even **Remove Node** is chosen on a particular node, the client components, if installed, will remain installed on the node. They will appear in Add or Remove Programs as **Microsoft SQL Server 2008**, and can be remove them from there.

## Considerations for Installing Other Components in a Failover Cluster

In addition to the SQL Server Database Engine, you can also install other components into a failover cluster, including SQL Server Analysis Services, SQL Server Reporting Services, and SQL Server Integration Services. There are special considerations for running these services within a failover cluster environment, which are outlined below. We will also address considerations for the Microsoft Distributed Transaction Coordinator (MSDTC).

### Analysis Services

Analysis Services is installed by SQL Server Setup as a generic application. It can be installed alongside a SQL Server instance, although this is not a requirement. By default, the Analysis Services resource is not dependent on the SQL Server resource.

When running on the cluster, Analysis Services listens on the default port – TCP 2383. Even if it is installed as a named instance, Analysis Services will detect that it is running on a cluster, and so it ignores the port server configuration property and listens on the default port. One of the implications of this behavior is that, when you have only Analysis Services installed on the cluster, you can disable the SQL Browser service because it is not needed to provide connectivity to Analysis Services. This also implies that all of the connections to Analysis Services that are running on a cluster should ignore the instance name that was given during installation and just use the Analysis Services’ network name. For example, when installed as ‘INST1’ into the cluster resource group ‘SQL1’, instead of connecting to ‘SQL1\INST1’, the connection to Analysis Services should use just ‘SQL1’.

**Note:** When working with a clustered instance of Analysis Services, you should not use instance rename functionality.

#### Analysis Services Deployment Best Practices

For best performance and minimal resource contention, it is recommended that Analysis Services instances do not share the same physical machine with other resource-intensive applications, including SQL Server. Production Analysis Services instances should be deployed to their own servers. For a failover cluster, this means that Analysis Services should be deployed to its own resource group and not installed into the SQL Server group.

Note that if all components are selected for installation during setup, Analysis Services is installed as a clustered resource into the SQL Server resource group by default. This may be adequate for lightly used systems or development systems, but is generally not recommended for production systems because SQL Server and Analysis Services will always be competing for system resources. Consider installing Analysis Services into a dedicated resource group with its own IP address, network name, and disk resources.

**Note:** SQL Server 2008 does not support adding features to an existing failover cluster instance, so Analysis Services cannot be added to an existing instance of SQL Server. To share a resource group with an instance of SQL Server, you must choose to install Analysis Services during the initial installation of SQL Server.

### Reporting Services

Reporting Services is not installed as a clustered resource. It is installed locally on the node, but is not configured by setup.

Reporting Services is most effectively deployed in a farm which provides both availability and load balancing. The backend SQL Server hosting the Reporting Services databases should still be clustered for maximum availability.

For more information about configuring Reporting Services for high availability, see the following topics in SQL Server Books Online:

* [Hosting a Report Server Database in a SQL Server Failover Cluster](http://technet.microsoft.com/en-us/library/bb630402.aspx) (http://technet.microsoft.com/en-us/library/bb630402.aspx)
* [Configuring Reporting Services for Scale-Out Deployment](http://technet.microsoft.com/en-us/library/ms156453.aspx) (http://technet.microsoft.com/en-us/library/ms156453.aspx)

### SQL Server Integration Services

SQL Server Integration Services is not installed as a clustered resource. Each physical node of a cluster may only have one instance of Integration Services. While it is possible to configure Integration Services as a clustered resource, the recommended best practice is to install it locally on all nodes of the cluster instead. Integration Services packages for a given instance can then be run via SQL Server Agent, which is cluster-aware. Packages should be created so that they can be restarted and be resumed after a failover. For more information, see:

* [Configuring Integration Services in a Clustered Environment](http://msdn2.microsoft.com/en-us/library/ms345193.aspx) (http://msdn2.microsoft.com/en-us/library/ms345193.aspx)
* [Description of the SQL Server Integration Services (SSIS) service and of alternatives to clustering the SSIS service](http://support.microsoft.com/default.aspx?scid=kb;EN-US;942176) (http://support.microsoft.com/default.aspx?scid=kb;EN-US;942176)

### Microsoft Distributed Transaction Coordinator

Some SQL Server implementations will require the installation of the Microsoft Distributed Transaction Coordinator (MSDTC). MSDTC is a component of Windows Server and is leveraged by some SQL Server features and applications to coordinate transactions across multiple servers.

On Windows Server 2003, there can be only one clustered MSDTC resource per cluster. This means that all applications across the entire cluster will are mapped to a single instance of MSDTC. It is recommended that you install this MSDTC instance in its own resource group to increase overall availability and minimize the impact of an MSDTC failure. For more information about setting up MSDTC on a Windows Server 2003 failover cluster, see [How To Configure MSDTC on a Windows Server 2003 Cluster](http://support.microsoft.com/kb/301600/) (http://support.microsoft.com/kb/301600/).

On Windows Server 2008, you can install multiple instances of MSDTC on a single failover cluster. The first instance of MSDTC that is installed will be the cluster default instance of MSDTC. This can be changed via the Component Services Management Console (dcomcnfg). To modify this, perform the following actions:

1. On the **Start** menu, click **Run**, type **dcomcnfg** and then press ENTER to launch the Component Services Management Console.
2. Expand **Computers**, and then right-click **My Computer**.
3. Click **Properties**, click the **MSDTC** tab, and then select the default coordinator for your cluster.

SQL Server 2008 will take advantage of an instance of MSDTC installed to the SQL Server’s local cluster resource group by automatically using the instance of MSDTC. However, individual applications can be mapped to any instance of MSDTC on the cluster. For more information about this, see “Mapping an Instance of SQL Server 2008 to an Instance of MSDTC” below. Figure 1 shows the rules that apply for an instance of MSDTC to be chosen on SQL Server 2008.

**Figure 1:** Process flow for MSDTC mapping

**Note:** If the MSDTC instance that is installed to the local cluster group of SQL Server fails, SQL Server does not automatically attempt to use the default cluster instance or the local machine instance of MSDTC. You would need to completely remove the failed instance of MSDTC from the SQL Server group to use another instance of MSDTC. Likewise, if you create a mapping for SQL Server and the mapped instance of MSDTC fails, your distributed transactions will also fail. If you want SQL Server to use a different instance of MSDTC, you must either add an instance of MSDTC to the local cluster group of the SQL Server or delete the mapping.

For more information about creating an MSDTC resource for Windows Server 2008 failover clustering, see [Checklist: Creating an MS DTC Resource in a Windows Server 2008 Failover Cluster](http://technet.microsoft.com/en-us/library/cc725955.aspx) (http://technet.microsoft.com/en-us/library/cc725955.aspx).

#### Enabling Network Access for MSDTC

After creating the MSDTC resource(s) for your cluster, you must enable network access for the MSDTC resource. This is done from the Component Services Management Console. . In the Component Services Management Console, perform the following actions:

1. Expand **Component Services**, expand **Computers**, expand **My Computer**, expand **Distributed Transaction Coordinator**, and then expand *<Your instance of MSDTC>*.
2. Right-click the instance that you want to configure, and then click **properties**.
3. Under **Security Settings**, select the **Network DTC Access**, **Allow Inbound**, and **Allow Outbound** check boxes, and then click **OK** to complete the configuration.

These configuration options enable MSDTC to access resources on the network and allow applications to access this instance of MSDTC from remote machines including other cluster nodes. It is not necessary to perform these actions on each node of the cluster because the changes will propagate to all nodes of the cluster for a clustered instance of MSDTC.

#### Mapping an Instance of SQL Server 2008 to an Instance of MSDTC

To map a SQL Server instance to an instance of MSDTC, open an administrative command prompt and use the msdtc.exe command.

Create a mapping

The following command is used to create a mapping between an instance of SQL Server and an instance of MSDTC.

msdtc -tmMappingSet -name <MappingName> -service <SQLServerServiceName> -clusterResource <MSDTCResourceName>

<MappingName> is an arbitrary name that you choose to identify this mapping.

<SQLServerServiceName> is the service name for your SQL Server instance. If this is the default instance, use MSSQLServer as your service name. If this is a named instance, use MSSQL$<InstanceName> as your service name.

<MSDTCResourceName> is the resource name for the instance of MSDTC to which you want to map SQL Server.

**Note:** If you create an incorrect mapping, the MSDTC command still succeeds, but your mapping will not work correctly.

View existing mappings across the entire cluster

The following command is used to view all MSDTC mappings across the entire cluster.

msdtc -tmMappingView \*

Delete an existing mapping

The following command is used to delete and existing MSDTC mapping.

msdtc -tmMappingClear -name <MappingName>

Test access by executing BEGIN DISTRIBUTED TRAN from a SQL Server Management Studio query window. If it fails to enlist the transaction, you can retrieve the error from the SQL Server error log. An error of XACT\_E\_CONNECTION\_DOWN at this point generally indicates that your MSDTC instance is not running, your mapping is incorrect, or you are being blocked by a firewall.

#### Best Practices for MSDTC Configuration with SQL Server 2008

Generally, dedicated instances of MSDTC for each SQL Server or application instance that uses MSDTC regularly are recommended. This is because it prevents the failure of one instance of MSDTC from affecting distributed transactions for other applications. However, when configuring MSDTC for an instance of SQL Server 2008, the specific needs of your environment—in terms of performance, availability, or manageability—will determine the best practice for your MSDTC configuration.

The most common MSDTC configurations are outlined below along with their benefits and other considerations:

MSDTC in Each SQL Server Resource Group

* This option provides the best performance. It guarantees that your instance of MSDTC will always run on the same physical node as the SQL Server, which reduces communications overhead.
* It is suitable for the widest array of requirements.
* It does not require mappings . SQL Server 2008 automatically uses the local cluster group MSDTC resource by default. Note that if this local resource is offline or failed, distributed transactions will fail for this instance of SQL Server, and you will need to delete or move the MSDTC resource to make use of another instance of MSDTC on the cluster.
* This configuration conserves drive letters for clusters that have a limited number of drive letters available. You can make use of mount points to keep disk I/O separate from other SQL Server files without using up a drive letter on the cluster.

**Note:** When you use this configuration, you must determine the correct setting for the **If restart is unsuccessful, fail over all resources in this service or application** setting of the MSDTC resource. If the function of MSDTC is critical to your environment, you should either set MSDTC to **If restart is unsuccessful, fail over all resources in this service or application** or put MSDTC in its own resource group and create a mapping that directs SQL Server to that instance of MSDTC. In most cases, you will not want to set **If restart is unsuccessful, fail over all resources in this service or application**. The setting enables MSDTC to be restarted on the same node if it fails, but if it cannot be restarted, it will not cause a failover of the entire SQL Server resource group.

Dedicated Group for Each MSDTC Instance

* This configuration is recommended only if the primary concern is maximum availability of SQL Server and MSDTC and if the performance of remote MSDTC has been verified to be acceptable for the application and load.
* MSDTC requires a dedicated physical disk resource in this configuration. If the cluster hosts many instances of SQL Server, you may not have enough disk resources to give every instance of SQL Server a dedicated MSDTC in a separate resource group.
* Manageability can be a challenge if you have many SQL Server-to-MSDTC mappings.

**Note:** The performance of distributed transactions might be better when SQL Server and the MSDTC resource to which it is mapped are running on the same physical node. Be sure to consider this factor when testing application performance, and test with MSDTC on remote nodes.

Single Default Cluster MSDTC

* Windows Server 2003 failover clusters can only have a single clustered instance of MSDTC. It is recommended that you install this instance to a dedicated resource group. This configuration prevents failures of MSDTC from affecting other applications.
* On Windows Server 2008 failover clusters, we recommended that you have the default MSDTC instance installed to a dedicated resource group. This default instance will provide MSDTC services for all applications that are not specifically mapped to another instance.
* SQL Server instances that rarely use MSDTC can use the default MSDTC instance on an ad-hoc basis.

Local MSDTC Instance

* Each node of the cluster also hosts a local instance of MSDTC.
* This option provides no high availability for MSDTC in the event of a failure, because the MSDTC log is stored on a local disk instead of a cluster disk.
* Even though no mappings exist for the local MSDTC instance, it is still required to have the local MSDTC instance running. During application/MSDTC initialization, the local MSDTC instance is used to determine the identity of the correct instance of MSDTC for this application (local, cluster default, or a specifically mapped instance). It is also used for non-cluster-aware applications on this node.
* This is the least desirable option for enabling distributed transactions on a failover cluster, and it is not recommended for high availability applications.

It is acceptable to create a hybrid of the above configurations to meet the needs of your environment.

# SQL Server Upgrade from Previous Versions

There are several things to consider when you prepare for an upgrade to a SQL Server 2008 failover cluster from SQL Server 2000 or SQL Server 2005. See “[SQL Server 2008 Upgrade Technical Reference Guide](http://www.microsoft.com/downloads/details.aspx?familyid=66d3e6f5-6902-4fdd-af75-9975aea5bea7&displaylang=en&tm)” (http://www.microsoft.com/downloads/details.aspx?familyid=66d3e6f5-6902-4fdd-af75-9975aea5bea7&displaylang=en&tm) for a comprehensive discussion of the upgrade, with specific details about upgrading a SQL Server failover cluster.

You can find further discussion in “[How to: Upgrade a SQL Server Failover Cluster Instance (Setup)](http://msdn.microsoft.com/en-us/library/ms191295.aspx)” (http://msdn.microsoft.com/en-us/library/ms191295.aspx) in SQL Server 2008 Books Online*.*

## Reducing Downtime by Using Rolling Updates and Upgrades

In SQL Server 2005 and earlier versions, applying an upgrade, service pack, or cumulative update required taking the entire SQL Server instance down during the installation process. This resulted in reduced availability, depending on the duration of the installation, across all failover cluster nodes that could host the SQL Server instance.

To improve availability during installations, SQL Server 2008 failover clustering introduced rolling upgrades and updates. A rolling upgrade refers to a version upgrade, for example, from SQL Server 2005 to SQL Server 2008. A rolling update applies to the installation of a service pack (for example, SQL Server 2008 with no service pack to SP1) or a cumulative update. A rolling upgrade or update means that you can apply changes to the passive nodes of a failover cluster, fail over the SQL Server instance that has not updated to the updated node, and then apply the upgrade or update to remaining nodes. The total outage time for the SQL Server instance is the amount of time that the SQL Server instance takes to fail over to the updated node and apply the metadata changes for the update. Upon failover to the newly updated node, the SQL Server instance is automatically upgraded and updated with unattended scripts that run against the database engine. You must then update any passive nodes that have not yet been updated to ensure that versions of binaries are identical across the failover cluster nodes.

Please note that, although rolling upgrades increase the overall availability of a SQL Server instance, they are not required. You may still apply updates directly on the active node that is hosting the SQL Server instance, but this will result in longer downtime for the SQL Server instance.

The next two sections explain how to implement rolling upgrades and updates.

### Rolling Upgrades

**Note:** This section describes the process behind performing rolling upgrades from a previous version of SQL Server. Keep in mind that this section discusses upgrades for SQL Server failover clustered instances and does not apply to upgrades for stand-alone SQL Server instances that are not failover clustered. Also, note that an “in-place” upgrade path for a stand-alone instance to a failover cluster is not available. Your only available option in that scenario is to perform a side-by-side database migration from the stand-alone instance to the SQL Server 2008 failover cluster instance.

Performing a successful rolling upgrade requires that you upgrade passive nodes one at a time, initiate a failover of the SQL Server instance to the upgraded node, and lastly upgrade any failover cluster nodes that have not been upgraded yet. For rolling upgrades, part of the “rolling process” is automatic and the setup installation process itself handles it. Rolling upgrades have some procedural considerations and steps that may be required prior to execution. These are outlined below.

To begin the rolling upgrade process, start by installing and updating the prerequisites, listed below, on each passive node of the failover cluster. The number of nodes in your failover cluster, in addition to the number of SQL Server instances, determines the number of failovers you may need to perform to minimize the cumulative downtime.

The prerequisite steps that are required to minimize upgrade downtime are as follows:

* Install the Microsoft .NET Framework 3.5 SP1 (except for Windows Server 2003 IA64, for which you need to install the .NET Framework 2.0 SP2), and Windows Installer 4.5 on all nodes before you begin the upgrade. Installing these components typically requires a restart of each node. Therefore, to minimize downtime, consider installing these components and restarting the nonactive nodes first. Then, fail over the SQL Server instance to a passive node before you install these components on what was the active node.
* Upgrade the shared components on each node.
* If you are upgrading SQL Server on Windows Server 2003 SP2, you must install the hotfix that is referenced in “[Error message when you try to create a cluster file share resource in a Windows Server 2003-based cluster: “System error 87 has occurred (0X00000057)](http://support.microsoft.com/kb/937444)” on each node in the failover cluster. This hotfix requires a restart after installation.

The number of nodes in the failover cluster, in addition to the number of SQL Server instances that are hosted on the failover cluster, affects how you can maximize the availability of the SQL Server instance during the rolling upgrade process. For example, a two-node failover cluster has limited options for maintaining high availability during an upgrade. This is because, if an unplanned failover was required from the active and nonupdated node, there are no additional nodes on which to host the SQL Server instance, so high availability is at risk. Alternatively, if you have a four-node failover cluster, a rolling update can be performed in pairs and you can still maintain high availability (by upgrading two nodes at a time). Unlike rolling updates, which this paper describes in detail later, the upgrade process has some built-in logic that helps it to maximize availability.

When you upgrade a failover cluster instance by using the GUI installation, the setup automatically determines the point at which to fail over the SQL Server instance to an upgraded node. This automatic procedure is based on the number of nodes in the failover cluster, taking into account which nodes are upgraded and which nodes are not. This helps to maximize high availability during the upgrade. After half or more of the nodes are upgraded, the setup process initiates a failover to the upgraded node by default when you perform an upgrade on the next node that has not been upgraded. Setup manages the possible owners of the SQL Server instance, removes nodes that have not been upgraded, and adds upgraded nodes as you update each node in the failover cluster.

If you do not want the setup process to automatically control the upgrade process, you can upgrade your SQL Server instance by using command-line calls in conjunction with the */FAILOVERCLUSTERROLLOWNERSHIP* parameter. This parameter has three different options:

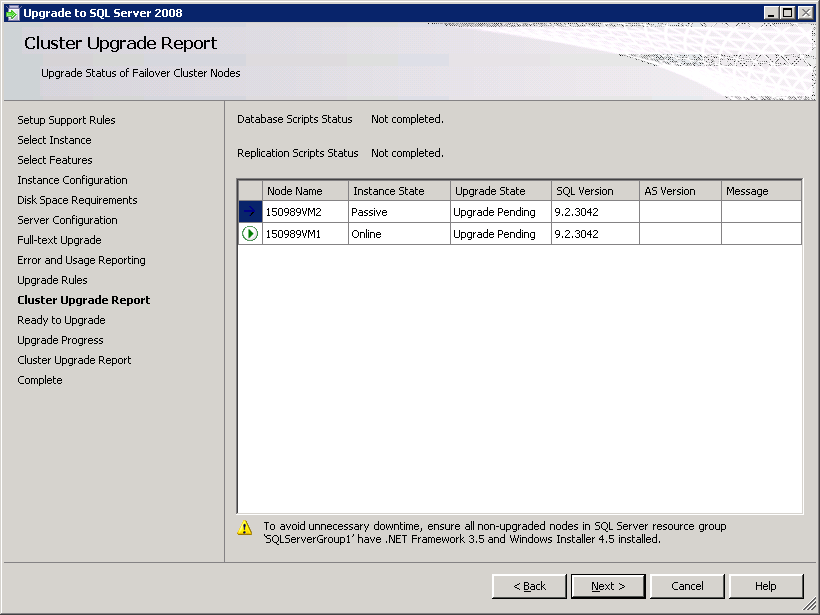
* /FAILOVERCLUSTERROLLOWNERSHIP = 0. Designating this option means that SQL Server instance cluster ownership is not failed over to upgraded nodes. After the node is upgraded, the current node is not added as a possible owner to the SQL Server instance.
* /FAILOVERCLUSTERROLLOWNERSHIP = 1. This option specifies first to fail over the instance from the current active node that is being upgraded; second, to upgrade the node; and finally, to add the upgraded node as a possible owner.
* /FAILOVERCLUSTERROLLOWNERSHIP = 2. This is the default setting, which you will encounter when you use the attended (GUI-based) upgrade installation. It results in automatic management of cluster ownership depending on the number of nodes and their upgrade state. If less than half of the nodes have been upgraded, the */FAILOVERCLUSTERROLLOWNERSHIP* parameter follows the behavior of option 0 (cluster ownership is not rolled to upgraded node and node ownership is not added). If half or more than half of the nodes have already been upgraded, the */FAILOVERCLUSTERROLLOWNERSHIP* parameter follows the behavior of option 1 (SQL Server instance is failed over to an upgraded node and the current upgraded node is added as an owner after the node upgrade is complete).

Figure 2 details the general rolling upgrade process flow.

**Figure 2:** The process flow for a SQL Server failover cluster rolling upgrade

If multiple SQL Server instances are hosted across nodes, you will need to do some rearranging of ownership before installation to minimize downtime and keep the process manageable.

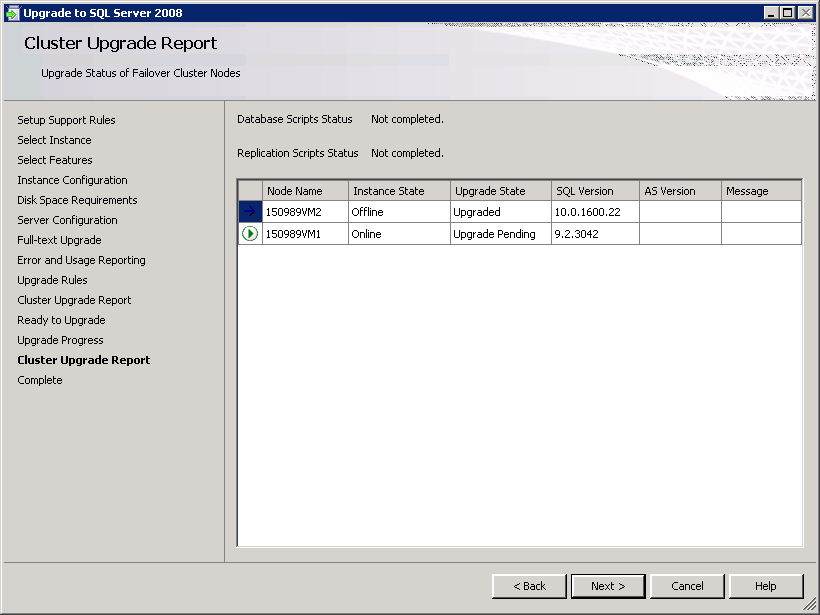
When you perform the upgrade installation on a passive node, you should expect to see information in the Cluster Upgrade Report that references which nodes are active and passive. In Figure 3, you can see that both nodes show an Upgrade State of Upgrade Pending, and also see which node is passive and which is online.



**Figure 3:** The Cluster Upgrade Report in the **Upgrade to SQL Server 2008** dialog box

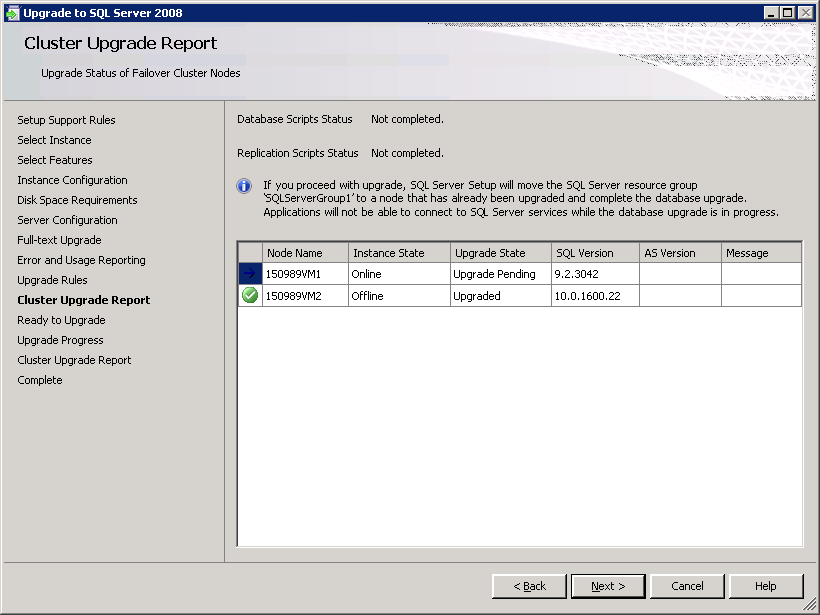
In Figure 3, note the warning message at the bottom of the screen, which advises that you should install the .NET Framework 3.5 and Windows Installer 4.5 to avoid unnecessary downtime.

After you have completed the upgrade of the passive node on a two-node failover cluster, the Cluster Upgrade Report in the **Upgrade to SQL Server 2008** dialog box appears again and shows the updated upgrade state (Figure 4).



**Figure 4:** The Cluster Upgrade Report in the **Upgrade to SQL Server 2008** dialog box after an upgrade

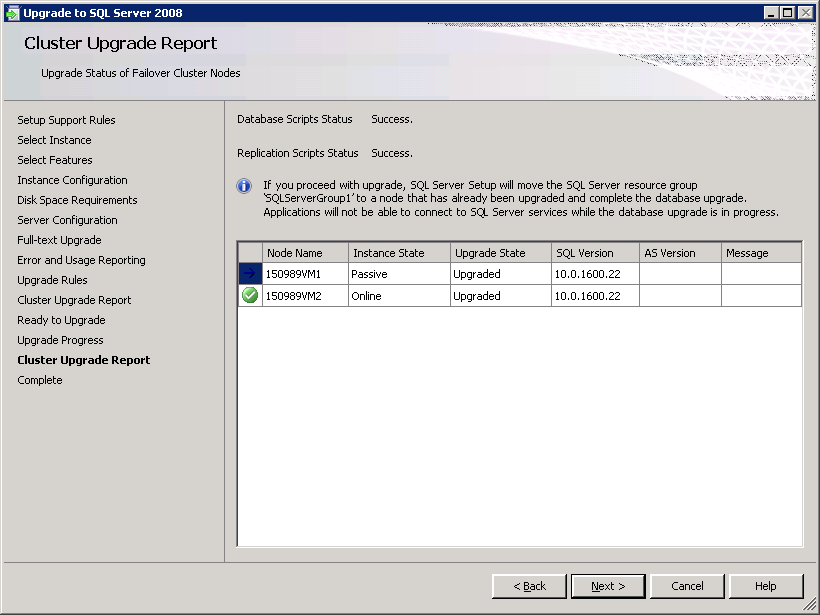
During the upgrade of the *active* node, you should expect to see an upgrade status that resembles Figure 5.



**Figure 5:** The Cluster Upgrade Report in the **Upgrade to SQL Server 2008** dialog box for an active node

Note the information message above the table. The actual outage period for the SQL Server instance is the amount of time that it takes to fail over the SQL Server instance and then apply the database upgrade scripts.

Upon successful upgrade of the final node, the Cluster Upgrade Report should look similar to Figure 6.



**Figure 6:** The Cluster Upgrade Report in the **Upgrade to SQL Server 2008** dialog box after upgrading the active node

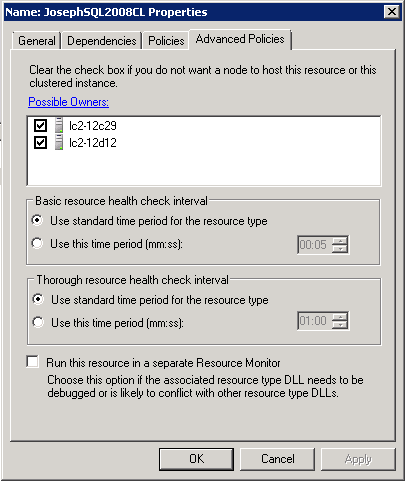
Both SQL Server versions should match after the final node upgrade. After all nodes are upgraded, in the Failover Cluster Management tool, in the SQL Server network resource, on the **Advanced Policies** tab, you should confirm that all appropriate owners are checked. After validating this, you should also test fail over of the SQL Server instance across all permitted nodes to identify any post-upgrade failover cluster issues.

### Rolling Updates

Rolling updates refer to the installation of cumulative updates and service packs on a SQL Server 2008 instance. Rolling updates are intended to minimize downtime and maximize availability during the update process. This section describes the key concepts and steps for performing a rolling update, but you can also find step-by-step instructions for applying updates to a SQL Server instance in “[SQL Server 2008 failover cluster rolling patch and service pack process](http://support.microsoft.com/kb/958734)” (http://support.microsoft.com/kb/958734).

Unlike rolling upgrades, rolling updates do not contain embedded logic to determine instance failover. Therefore, the person who installs an update is responsible for maximizing availability through proper planning and execution. Before you perform a rolling update, the first consideration is to determine which nodes you can use to host the SQL Server instance on the failover cluster. This is an important factor to avoid mixed versions across nodes for the SQL Server instance.

For example, if you are performing an update on a passive node and your SQL Server instance on the active node fails over while you are performing the upgrade, you could encounter update failures and corruption issues. To mitigate the risk of unplanned failovers, you should evaluate the number of nodes that can host the SQL Server 2008 instance. Then, you should remove the passive nodes that you plan to update first from the list of possible owners on the SQL Server instance network name resource. You can configure this by using the **Advanced Policies** tab, as Figure 7 shows.



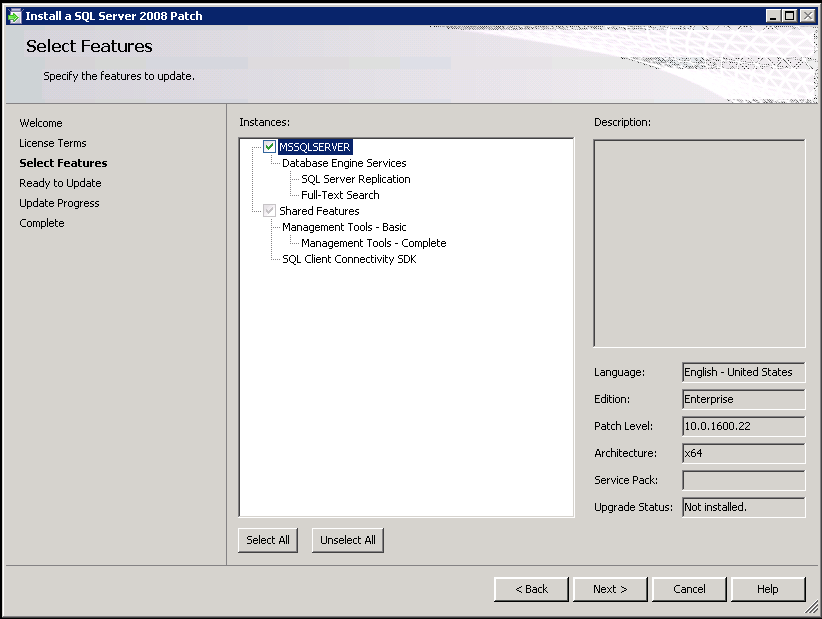
**Figure 7:** The **Advanced Policies** tab for a SQL Server network resource

In Figure 7, notice that two servers are currently selected as possible owners of the network name resource. If you planned to update the lc2-12d12 passive node first, you would need to make sure that the SQL Server instance was currently hosted on lc2-12c29 and then clear the check box for this server before you performed the update. Modifying the list of available nodes does not require an outage of the resource. Also, note that you must modify the SQL Server network name resource for node ownership because it is the resource that is validated for possible node owners by the failover cluster service.

If more than two nodes in your failover cluster can host a SQL Server instance, you may consider keeping at least two possible owners selected at any given time during the update process. This enables you to maintain high availability with at least two nodes during an update of the remaining passive nodes.

First, you should determine which nodes can host the SQL Server instance and then remove resource ownership for the passive hosts that you wish to update. Then, you should move any other active failover cluster applications and services to other nodes in case the update (service pack or cumulative update) requires a restart of the node. After you have ensured that the node that you are updating is truly passive, you can proceed with applying the update.

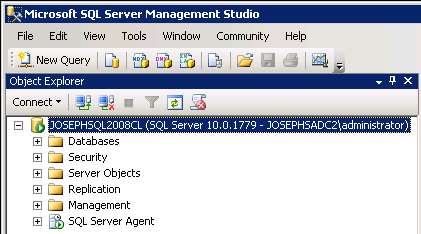
After you initiate the installation of an update on a passive node, you will have a chance to validate the current node version numbers on the **Select Features** page of the **Install a SQL Server 2008 Patch** dialog box. Figure 8 shows the selected features for a SQL Server 2008 cumulative update. Notice that, although this is being executed on a passive node, the cumulative update setup still recognizes the various engine components that are available to be upgraded.



**Figure 8:** The **Select Features** page of the **Install a SQL Server 2008 Patch** dialog box for a cumulative update

After you have updated the first passive node, you can repeat the same installation steps if other passive nodes form part of your first round of planned updates. For example, if you have a four-node cluster, and you are updating two passive nodes, you would repeat the update process for each of the two passive nodes that were removed as potential hosts of the SQL Server instance.

After you have completed the updates to the passive nodes, in the Failover Cluster Management tool, in the SQL Server network resource, on the **Advanced Policies** tab, you should reselect the updated nodes to enable hosting of the SQL Server instance. After you have re-enabled the updated nodes, you can then fail over the SQL Server instance from the node that has not been updated to an updated node. After failover to the updated node, update scripts automatically run against the engine. You can use the Failover Cluster Management tool to confirm that the SQL Server instance resources came online on the newly updated node. Then, you can confirm the version number of the SQL Server instance by connecting to the SQL Server instance by using SQL Server Management Studio and opening Object Explorer, as shown in Figure 9.

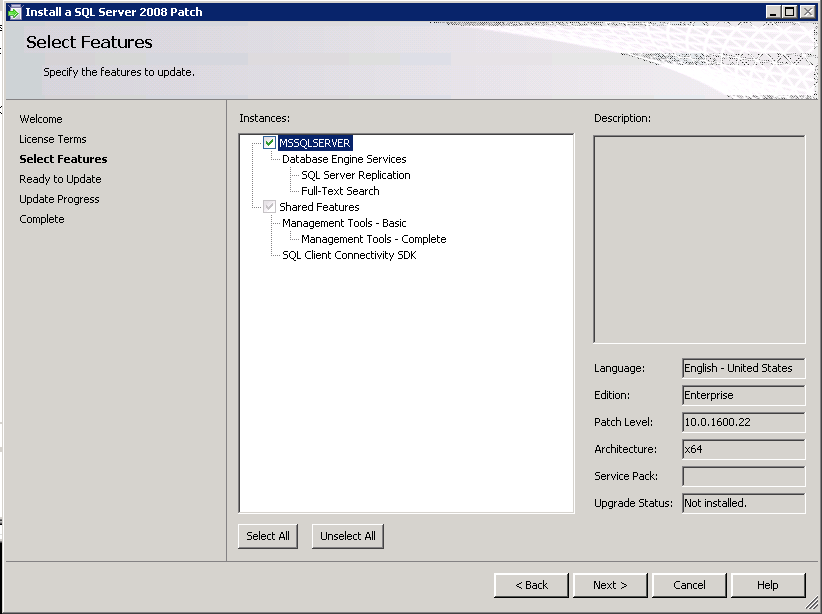


**Figure 9:** The Object Explorer window in SQL Server Management Studio

In Figure 9, the version of the failed-over SQL Server instance is 10.0.1779, which represents the Cumulative Update Package 2 for SQL Server 2008. Before failover from the node that has not been updated, the version number for the SQL Server instance was 10.00.1600.22 (RTM). After failover, you should check the SQL Server error log for any error messages that are related to the update.

After you have failed over the SQL Server instance to an updated node, you should then update the remaining nodes that have not been updated. As before, there is a risk of failover of the updated SQL Server instance to a down-level node during your installation. Therefore, in the Failover Cluster Management tool, in the SQL Server network resource, on the **Advanced Policies** tab, you should clear the passive nodes that have not been updated until all of them have been updated.

When you install the update on the remaining passive nodes, the previous SQL Server engine versions are displayed. Figure 10 shows an example of this, with the node that has not been updated displaying an older engine version of 10.0.1600.22 in the **Patch Level** box.



**Figure 10:** The **Select Features** page of the **Install a SQL Server 2008 Patch** dialog box for a node that has not been updated

After you have updated each node, in the Failover Cluster Management tool, in the SQL Server network resource, on the **Advanced Policies** tab, you should re-enable all updated nodes to allow hosting of the SQL Server instance. You should also perform a failover of the SQL Server instance across updated nodes to validate the installation and ensure full availability of the resources.

Figure 11 shows the general process flow for performing a rolling update.

**Figure 11:** The process flow for a SQL Server failover cluster rolling update

In summary, rolling updates are performed on passive nodes first to minimize the downtime of the SQL Server instance. Unlike rolling upgrades, the update process does not include automated workflow logic that controls how the rolling upgrade is managed for availability. Therefore, you must plan your update process carefully to ensure maximum availability and avoid inadvertent up-level or down-level SQL Server instance failovers during the upgrade process.

# Administration and Maintenance of a SQL Server 2008 Failover Cluster

There are several administrative and maintenance considerations that, in previous versions, required different steps for a failover cluster instance when compared to a stand-alone SQL Server instance. Using SQL Server 2008, many of these tasks have the same steps regardless of the environment (failover cluster or stand-alone). The following section discusses some of these tasks.

## Changing the Network Name of a SQL Server Failover Cluster

Both SQL Server 2005 and SQL Server 2008 failover clusters support changing the SQL Server failover cluster network name after installation. This was not possible with SQL Server 2000 failover clusters. A SQL Server failover cluster name can contain the network name and an instance name. For a default instance, there is a network name for the failover cluster instance, but no instance name. For example, if the SQL Server failover cluster was called VS1\instance1, SQL Server 2008 supports changes to the VS1 part of the name. SQL Server 2008 does not support changing the instance name (instance1) without a reinstallation of SQL Server.

For further information about renaming a SQL Server failover cluster, and step-by-step instructions, see “[How to: Rename a SQL Server Failover Cluster Instance](http://msdn.microsoft.com/en-us/library/ms178083.aspx)” (http://msdn.microsoft.com/en-us/library/ms178083.aspx) in SQL Server 2008 Books Online.

## Changing the IP Address of a SQL Server Failover Cluster

You can change or modify an IP address that the SQL Server failover cluster has been configured to use. See “[How to: Change the IP Address of a SQL Server Failover Cluster](http://msdn.microsoft.com/en-us/library/ms190460.aspx)” (http://msdn.microsoft.com/en-us/library/ms190460.aspx) in SQL Server 2008 Books Online for the steps to do this.

## Changing the Service Accounts for a SQL Server Failover Cluster

The following steps are similar to what you would need to do on a stand-alone instance of SQL Server to change the service accounts. You should use SQL Server Configuration Manager to change the service accounts. Bear in mind the following points when you manage the service accounts for a failover cluster:

* Do not change passwords for any of the SQL Server service accounts when a failover cluster node is down or offline. If you must do this, you must reset the password again by using SQL Server Configuration Manager when all nodes are back online.
* If the service account for SQL Server does not have administrative rights on the Windows Server cluster, you cannot delete the administrative shares on any nodes of the cluster. The administrative shares must be available on a cluster for SQL Server to function.

For a more in-depth discussion about using Windows accounts with SQL Server, see “[Setting Up Windows Service Accounts](http://msdn.microsoft.com/en-us/library/ms143504.aspx)” (http://msdn.microsoft.com/en-us/library/ms143504.aspx) in SQL Server 2008 Books Online. For recommendations about changing service accounts in a failover cluster, see “[Maintaining a Failover Cluster](http://technet.microsoft.com/en-us/library/ms178061.aspx)” (http://technet.microsoft.com/en-us/library/ms178061.aspx) in SQL Server 2008 Books Online.

## Managing SQL Server Resources by Using SQL Server Management Studio and Cluster Administrator

SQL Server Management Studio is cluster-aware in SQL Server 2008 and SQL Server 2005. Many operations that you can perform by using SQL Server Management Studio, such as stopping and starting services on instances that are not clustered, are also supported for SQL Server failover clusters. However, there are several important SQL Server resource management tasks, such as checking the properties of any of the SQL Server resources, which you cannot achieve by using SQL Server Management Studio. Instead, you must use the Cluster Administrator tool in Windows Server 2003 or the Failover Cluster Management tool in Windows Server 2008 to perform these actions. There are some tasks, such as starting and stopping SQL Server services, which you can perform by using either SQL Server Management Studio or Cluster Administrator.

**Note**: Do not use **Services** in Control Panel to start or stop SQL Server 2008 services in a failover cluster, because **Services** is not cluster-aware.

For more information about using SQL Server tools to manage failover clusters, see “[Using SQL Server Tools with Failover Clustering](http://technet.microsoft.com/en-us/library/ms175549.aspx)” (http://technet.microsoft.com/en-us/library/ms175549.aspx) in SQL Server 2008 Books Online.

## Rebuilding System Databases in a SQL Server Failover Cluster

To rebuild the system databases for a SQL Server 2008 failover cluster, you must execute setup.exe with specific command-line switches. Although the process is similar to SQL Server 2005, the parameters have changed for SQL Server 2008. The rebuilding process, whether it is on a failover cluster instance or a stand-alone instance, is the same in SQL Server 2008.

A new feature of SQL Server 2008 is that the system database files that are used to rebuild the current system databases do not come from the original installation media. This enhancement enables you to rebuild system databases without needing to access the original installation media or DVD. The files are now located on any cluster node on which the instance of SQL Server is installed, in the BINN\templates directory. This directory contains the **master**, **model**, and **msdb** database and log files that were copied from the installation source as part of setup.

For a step-by-step process to rebuild system databases of a SQL Server 2008 failover cluster instance, and other considerations, see “[Rebuilding System Databases](http://msdn.microsoft.com/en-us/library/dd207003.aspx)” (http://msdn.microsoft.com/en-us/library/dd207003.aspx ) in SQL Server 2008 Books Online.

## Implementing Trace Flags

Trace flags are used to temporarily set specific server characteristics or to switch off a particular behavior. For example, trace flag 1204 enables deadlock reporting. For a list of documented trace flags, see “[Trace Flags (Transact-SQL)](http://msdn.microsoft.com/en-us/library/ms188396.aspx)” (http://msdn.microsoft.com/en-us/library/ms188396.aspx)in SQL Server 2008 Books Online. You should not use undocumented trace flags.

You can implement trace flags on a SQL Server 2008 failover cluster instance by using the **DBCC TRACEON** and **DBCC TRACEOFF** commands, or the -T startup option, in exactly the same way as you would on a stand-alone SQL Server 2008 instance.

The following code examples show how to use the **DBCC TRACEON** and **DBCC TRACEOFF** commands.

The following command sets trace flag 1204 globally.

DBCC TRACEON (1204,-1)

-1 sets the flag globally. To set a trace flag at the session level, omit the -1 argument.

The following command disables trace flag 1204 globally.

DBCC TRACEOFF (1204,-1)

The following command displays the status of all trace flags that are currently enabled globally.

DBCC TRACESTATUS(-1);

After you have enabled trace flags by using the **DBCC TRACEON** command, they remain enabled on the server until they are explicitly disabled by executing a **DBCC TRACEOFF** statement, or until the SQL Server instance is restarted. If you need trace flags to be enabled every time SQL Server is restarted, you can add the trace flags as a startup option by following these steps:

1. Open SQL Server Configuration Manager on the cluster node where the SQL Server 2008 failover clustering instance is running.
2. Click **SQL Server Services**, right-click the SQL Server service on which you want to add startup parameters, and then click the **Advanced** tab.
3. On the **Advanced** tab, in the **Startup Parameters** box, type the startup parameters that you require, separating each one with a semicolon, and then click **Apply**.
4. The following message appears: “Any changes made will be saved; however, they will not take effect until the service is stopped and restarted.”
5. Click **OK**.

You do not have to stop and restart the SQL Server service immediately. This is useful if, for example, you want to make a change now, but restart the service during a maintenance window or when no users are connected to the system. After you have restarted SQL Server, the trace flags will take effect regardless of the node on which SQL Server is running. It is recommended that you remove the trace flags after you have finished troubleshooting.

# Troubleshooting SQL Server Failover Clusters

An instance of SQL Server may experience an unplanned failover or a restart for various reasons. Most of these reasons are not specifically related to cluster issues, so, in these cases, you should follow standard SQL Server troubleshooting steps. When you determine that a problem is cluster-related, you can use the guidelines in this section to help you to troubleshoot your problem.

## Determining the Nature of the Failure

When you become aware of an outage, you should first simply define what actually went wrong. What resource failed? Was it the disk? The network? SQL Server? SQL Server Agent? A file share? Are you troubleshooting a single resource restart on the same node, or a resource group failover, or a failed resource? Was the SQL Server instance online the entire time but users could not connect? Does the problem occur on one node or all nodes? Understanding exactly which resource failed, on what node, and the actions that the cluster has performed to try to resolve the problem is critical to determining the root cause of the failure.

The Cluster service adds resource status messages in the cluster log and the system event log. The system event log is generally more user-friendly. You can usually identify the failed resource by looking for an event ID 1069 in the system event log.

If the system event log has no information about the cause of the failure (for example, the system stops writing events to the log), it will be necessary to use the cluster log. The cluster log is a bit more difficult to analyze, the log is more verbose, and the time stamps are in Coordinated Universal Time (Greenwich Mean Time) as opposed to the local system time. The “Windows Server Failover Clustering” section below provides an example, but an in-depth analysis of the cluster log is not within the scope of this white paper.

After the logs are collected, you should conduct a review of them, and understand the components that constitute the cluster, to ascertain the root cause of the problem.

## Root Cause Analysis

After trying to evaluate what failed and collecting the logs, the next step is to begin investigating the root cause of the failure. If you are still unsure of which component of the system is responsible for the problem, start from the lowest level (hardware), and work your way up through Windows Server and then SQL Server. If it seems obvious from your initial review of the logs which component was the source of the problem, you can start by troubleshooting that component. However, use caution when jumping to conclusions. Frequently, you will find that the resource that appeared to be the root cause was simply at the mercy of another component on the system that was failing or not performing at the level that it should.

### Hardware

Review the Windows Server system event logs and look for any hardware warning or error events around the time when the resource failure first occurred. At this point, it is not relevant which resource has failed; if you have a hardware problem near the time of the outage, you must address that first.

If you find any hardware errors, troubleshooting should continue as on a stand-alone server.

**Note:** Some disk subsystems produce errors or warnings during startup and shutdown, and fail over such (for example, a small computer system interface (SCSI) time-out error). If this is normal for your storage system, you can ignore these errors. Verify with your hardware vendor if you are unsure.

### Operating System and Network

Review Windows Server system and application event logs for any operating system or network errors. Examples include network interface card (NIC) failures, connectivity issues, system resource warnings, memory errors, Kerberos failures, and DNS registration errors. Compare driver versions, operating systems, and network configuration settings of nodes that are not working with nodes that are. Troubleshoot these problems from the level of the operating system.

Always verify network configuration. It is a best practice to validate the network configuration whenever you have a problem on the cluster. Incorrectly configured networking can cause sporadic and inconsistent behavior on the cluster. See “[Recommended private "Heartbeat" configuration on a cluster server](http://support.microsoft.com/kb/258750)” (http://support.microsoft.com/kb/258750) for specific details about proper configuration of cluster networks.

### Security

Review Windows Server system, application, and security logs for access being denied or other security errors. Security errors will manifest as problems in other components of the cluster, but the security event log may log valuable information.

### Windows Server Failover Clustering

Review Windows Server system and application event logs, and cluster logs. You can view cluster-related event log entries on a Windows Server 2008 cluster directly from the Failover Cluster Management MMC.

The following cluster log example illustrates a SQL Server instance that failed to come online and ultimately resulted in a failed SQL Server resource. In this particular case, the root cause of the problem was that a nondefault instance of SQL Server was configured to listen on a static TCP port with the SQL Browser service disabled and no alias configured on the node. This prevented the Cluster service from making a connection to the SQL Server instance to run the **IsAlive** check.

00000a98.00000fcc::2009/05/27-05:08:02.295 ERR [RES] SQL Server <SQL Server (SQLADV1)>: [sqsrvres] ODBC sqldriverconnect failed

00000a98.00000fcc::2009/05/27-05:08:02.295 ERR [RES] SQL Server <SQL Server (SQLADV1)>: [sqsrvres] checkODBCConnectError: sqlstate = HYT00; native error = 0; message = [Microsoft][SQL Server Native Client 10.0]Login timeout expired

00000a98.00000fcc::2009/05/27-05:08:02.295 ERR [RES] SQL Server <SQL Server (SQLADV1)>: [sqsrvres] ODBC sqldriverconnect failed

00000a98.00000fcc::2009/05/27-05:08:02.295 ERR [RES] SQL Server <SQL Server (SQLADV1)>: [sqsrvres] checkODBCConnectError: sqlstate = 08001; native error = ffffffff; message = [Microsoft][SQL Server Native Client 10.0]A network-related or instance-specific error has occurred while establishing a connection to SQL Server. Server is not found or not accessible. Check if instance name is correct and if SQL Server is configured to allow remote connections. For more information see SQL Server Books Online.

00000a98.00000aa4::2009/05/27-05:08:16.000 ERR [RHS] RhsCall::DeadlockMonitor: Call ONLINERESOURCE timed out for resource 'SQL Server (SQLADV1)'.

00000a98.00000aa4::2009/05/27-05:08:16.000 ERR [RHS] Resource SQL Server (SQLADV1) handling deadlock. Cleaning current operation.

00000638.00000714::2009/05/27-05:08:16.000 INFO [RCM] HandleMonitorReply: FAILURENOTIFICATION for 'SQL Server (SQLADV1)', gen(0) result 5018.

00000638.00000714::2009/05/27-05:08:16.000 INFO [RCM] TransitionToState(SQL Server (SQLADV1)) OnlinePending-->ProcessingFailure.

00000638.00000714::2009/05/27-05:08:16.000 ERR [RCM] rcm::RcmResource::HandleFailure: (SQL Server (SQLADV1))

00000638.00000714::2009/05/27-05:08:16.001 INFO [RCM] resource SQL Server (SQLADV1): failure count: 1, restartAction: 2.

00000638.00000714::2009/05/27-05:08:16.001 INFO [RCM] Will restart resource in 500 milliseconds.

00000638.00000714::2009/05/27-05:08:16.001 INFO [RCM] TransitionToState(SQL Server (SQLADV1)) ProcessingFailure-->[Terminating to DelayRestartingResource].

Notice that each entry in the cluster log has a particular format. The process ID and thread ID of the thread that issues the log entry are the first part of each line, and are separated by a period. The next part of the line is the system time in Coordinated Universal Time. The third part of the line is the level of the event. This can be an error (ERR), a warning (WARN), or an information message (INFO). Following the event level is the cluster component that generated the message. Messages that the SQL Server resource generates show as [RES]. The message that the component is logging follows at the end. Note that you can obtain the error description from the error number. For system error codes, you can obtain the error description by using the **net helpmsg** command with the error code.

### SQL Server

If you encounter a SQL Server resource failure, your next step is to investigate the SQL Server error logs to determine whether this was a SQL Server failure or an inability of the Cluster service to connect to the SQL Server instance.

Some SQL Server error conditions can prevent the Cluster service from connecting and running **IsAlive** check or getting a timely response from the SQL Server instance. Examples include memory errors such as SQL Server memory being paged out, Scheduler hangs, and I/O stalls. None of these types of errors are specific to failover clustering. However, the **IsAlive** check is subject to the same problems that affect performance that other users’ queries experience. You should handle this class of problem just as you would handle the same problem on a stand-alone server.

#### SQL Server Connectivity

If you suspect a problem with SQL Server network connectivity, start with a review of the recent SQL Server error logs:

* Are there any obvious network errors?
* If you are using Kerberos, make sure that the service principal names are correct in Active Directory and that SQL Server is registering them correctly. Look for messages that are similar to the following near the beginning of the SQL Server error log:

The SQL Server Network Interface library successfully registered the Service Principal Name (SPN) [ MSSQLSvc/<FQDN>:<InstanceName> ] for the SQL Server service.

The SQL Server Network Interface library successfully registered the Service Principal Name (SPN) [ MSSQLSvc/<FQDN>:<PortNumber> ] for the SQL Server service.

* Is SQL Server listening on the correct network libraries and is it using the correct IP address and pipe name? You should only modify the SQL Server network configuration by using the SQL Server Configuration Utility. Check for messages that are similar to the following near the beginning of the SQL Server error log and verify that IP addresses and pipe names are correct for this instance:

Server is listening on [ 10.10.10.54 <ipv4> 54794].

Server local connection provider is ready to accept connection on [ \\.\pipe\SQLLocal\InstanceName ].

Server named pipe provider is ready to accept connection on [ \\.\pipe\$$\SQL1\MSSQL$InstanceName\sql\query ].

* Are any aliases defined on any node of the cluster for this SQL Server instance? Consider removing these aliases temporarily to see whether you can connect with the aliases removed.
* Is the SQL Server Browser service running on all nodes of the cluster? Although it is not required for a default instance that uses a static port, it is required for consistent operation of a named instance.

#### SQL Server Failing on One Node

If SQL Server fails to come online on only one node of the cluster, you should focus on differences between the nodes:

* Did installation complete successfully on every node where this instance will run?
* Were any configuration changes made on this node that were not made on the others, such as manual registry changes or networking changes?
* Was a hotfix or service pack applied to this node but not the others (for example, SQL Server, an operating system, or Microsoft Data Access Component (MDAC))?
* Were hardware or driver changes made to this node?
* Are there security configuration differences between nodes? Changes to group policies? Are server principal names configured correctly for each node?

#### Other Errors

Failures of other resources are investigated differently depending on the resource that failed. For example, for SQL Agent, investigate the SQL Agent logs; for file shares, investigate the system and application event logs.

## Further Troubleshooting Techniques

### Reading and Extracting Event Logs and Cluster Logs

Reading by using the Event Viewer on the cluster itself is optimal. This is because some event information can be lost when viewing event logs on other computers that lack the resource information to populate event descriptions from the server. However, if you will be saving the logs for post-mortem analysis, exporting the event logs as text or \*.csv files enables you to read the descriptions in events for services and drivers that would otherwise be unavailable on a remote machine. You can import \*.csv files into Microsoft Office Excel® to enable filtering and sorting.

On Windows Server 2003, cluster logs are located in C:\Windows\Cluster by default. On Windows Server 2008, you can generate the cluster logs by using the code in the following example at the command prompt.

CLUSTER.EXE <cluster name> LOG /GEN /COPY:<destination path for log files>

### Comparing Registries Across Nodes

Starting with SQL Server 2008, all SQL Server registry checkpoints are set on the SQL Server network name resource of the SQL Server instance (before SQL Server 2008, checkpointed registry keys were set for the SQL Server resource itself). This change improves the reliability of the SQL Server instance during failover by replicating the registry keys before the SQL Server online event. This greatly reduces the chance that a registry modification by an application that is not cluster aware will cause a failure.

If the SQL Server resource group fails to come online on one node, but works fine on other nodes, compare cluster registry entries of checkpointed SQL Server registry keys across nodes. Registry entries that are not in sync across the available nodes might be different for the following reasons:

* Manual modification by an administrative user or admin level script (for example, a logon script or domain policy).
* Modification by tools that are not cluster-aware. The tools in SQL Server 2008 are all cluster-aware, but some third-party tools are not.
* Registry key modification by a cluster-aware tool but not yet applied by the checkpoint process. The key will be written at failover (in this case, no action is needed).

When you compare registry entries across nodes, you are looking for differences in keys, values, and permissions. Modifications from default values can cause unexpected results.

### Manually Changing a Checkpointed Key

**Important:** This section contains steps that tell you how to modify the registry. However, serious problems might occur if you modify the registry incorrectly. Therefore, make sure that you follow these steps carefully. For added protection, back up the registry before you modify it. Then, you can restore the registry if a problem occurs. For more information about how to back up and restore the registry, see [How to back up and restore the registry in Windows](http://support.microsoft.com/kb/322756/) (http://support.microsoft.com/kb/322756/).

SQL Server 2008 uses the Cluster service’s Checkpoint Manager to keep critical registry entries in sync across all nodes of the cluster. The Checkpoint Manager monitors the checkpointed registry keys on the active node for any changes.

SQL Server 2008 registers all checkpoints on the network name resource of the SQL Server instance. This enables registry keys to be repaired if bad settings are preventing SQL Server from coming online without the need to remove or edit checkpoints. To determine which keys are checkpointed on a cluster resource, or verify that the correct checkpoints exist, execute the code in the following example.

Cluster.exe . resource “SQL Network Name (SQL1)” /checkpoints

Default registry checkpoints for the SQL Server–related resources are listed in the following code example.

SQL Server Network Name

HKLM\Software\Microsoft\Microsoft SQL Server\MSSQL10.KATMAI1\Cluster

HKLM\Software\Microsoft\Microsoft SQL Server\MSSQL10.KATMAI1\MSSQLServer

HKLM\Software\Microsoft\Microsoft SQL Server\MSSQL10.KATMAI1\Replication

HKLM\Software\Microsoft\Microsoft SQL Server\MSSQL10.KATMAI1\Providers

HKLM\Software\Microsoft\Microsoft SQL Server\MSSQL10.KATMAI1\SQLServerSCP

HKLM\Software\Microsoft\Microsoft SQL Server\MSSQL10.KATMAI1\CPE

HKLM\Software\Microsoft\Microsoft SQL Server\MSSQL10.KATMAI1\SQLServerAgent

Analysis Services

HKLM\Software\Microsoft\Microsoft SQL Server\MSAS10.KATMAI1\Cluster

HKLM\Software\Microsoft\Microsoft SQL Server\MSAS10.KATMAI1\CPE

By default, there are no registry checkpoints for cluster disks, IP addresses, SQL Server Agent, file shares that FileStream uses, or the SQL Server instance itself.

After the registry checkpoints have been defined on a resource, the Checkpoint Manager maintains the registry keys as follows:

* When a new registry key is specified for the resource, the specified key is checkpointed.
* When the resource that is associated with the checkpointed key is brought online, the registry keys are updated with the previously checkpointed information.
* When the resource is brought offline, all of the checkpoints that are associated with this resource are saved.
* When the resource is online and changes are made to the registry key that is registered with the cluster server for replication, the Checkpoint Manager ensures that the updates are written to a checkpoint that is maintained in the cluster database.

Microsoft recommends that you perform all SQL Server configuration changes through the SQL Server Client Tools. However, if you must make manual changes to the checkpointed registry keys, make the changes from the active node while the SQL Server network name resource is online. If you manually update these registry keys while the SQL Server network name resource is offline, the changes may not be replicated or may be lost.

### Bringing SQL Server Online Outside Cluster Services

Sometimes a SQL Server instance will not come online due to problems that prevent the Cluster service from getting a successful **IsAlive** result from the SQL Server instance. In these cases, it might be valuable to bring the SQL Server instance online from a command prompt so that you can perform troubleshooting and emergency maintenance.

This technique involves bringing a SQL Server instance online outside the control of the Service Control Manager. The SQL Server resource will appear to be offline from the Failover Cluster Management console and the Cluster service will not be managing the SQL Server instance. No failover or restart of the SQL Server instance will occur in this state. After it is online, the SQL Server instance will be available to clients on the network.

This technique is typically used to bring up the SQL Server instance when configuration issues are causing SQL Server to fail. Microsoft recommends that you only use this technique for intermediate troubleshooting steps because automated failover of the SQL Server instance is not available in this state. The technique involves the following steps:

1. Bring all SQL Server disk resources online via the Failover Cluster Management console.
2. Bring all SQL Server IP address resources online via the Failover Cluster Management console.
3. Bring the SQL Server network name resource online via the Failover Cluster Management console.
4. From the Command Prompt window, navigate to your SQL Server directory and execute the following code:  
   sqlservr.exe –c –s<instance name>
5. Leave the Command Prompt window open and do not minimize it. Closing the window or logging off will stop the SQL Server.
6. To stop the server, press CTRL+C. You will be prompted for confirmation of shutdown.

### Properties for the SQL Server Resource

The SQL Server resource exposes some properties that may be useful for more in-depth troubleshooting.

#### VirtualServerName

Always set this property to the virtual server name (also referred to as the network name) of the SQL Server instance. Do not change this value.

#### InstanceName

Always set this property to the instance name of the SQL Server instance. Do not change this value.

#### VerboseLogging

The default value for this property is 0 = off. Setting this value to 1 writes information events to the cluster log. This is generally not needed for SQL cluster troubleshooting, but does make it easier to determine the order of events and precise points of failure.

#### SQLDumperDumpFlags, SQLDumperDumpPath, and SQLDumperDumpTimeout

These properties enable and manage the Sqldumper.exe utility, which generates dump files. These settings are managed by the cluster command. For more information, see [How to use the Sqldumper.exe utility to generate a dump file in SQL Server 2005](http://support.microsoft.com/kb/917825) (http://support.microsoft.com/kb/917825).

The following code shows an example batch for configuring a mini-dump of all threads on failover with a time-out of 15 seconds for the dump to complete.

Cluster resource “SQL Server” /priv sqldumperdumpflags=0x0120

Cluster resource “SQL Server” /priv sqldumperdumppath=”path for dump files”

Cluster resource “SQL Server” /priv SQLDumperDumpTimeout=15000

To reset the properties to the default, execute the following batch.

Cluster resource “SQL Server” /priv:sqldumperdumpflags /usedefault

Cluster resource “SQL Server” /priv:sqldumperdumppath /usedefault

Cluster resource “SQL Server” /priv:SQLDumperDumpTimeout /usedefault

#### SQLDMVScriptTimeout

This property is not currently implemented. The default value is 0 = off.

#### SqlPreStartupActionsFlags

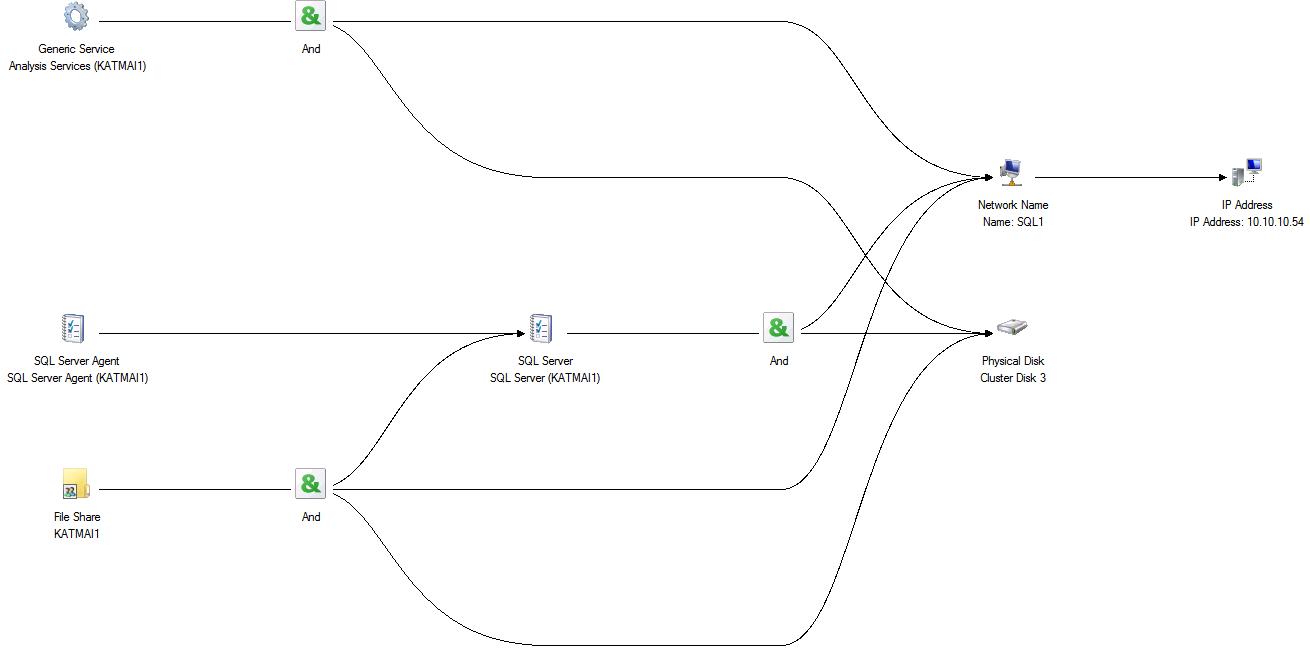
The default value for this property is 0 = off. If it is set to 1, it does not run any upgrade steps at all. Do not change the default value.

## Best Practices to Prevent Unnecessary Outages

### Evaluate Resource Dependencies

Evaluate dependencies for each cluster resource in your SQL Server resource group. Do not add additional dependencies unless they are truly needed. For example, adding a new disk resource to be used by a mission critical database warrants adding a dependency on this drive to the SQL Server resource. Unnecessary dependencies can slow down failover (waiting for a resource to come online when there is no need to do so) and cause unnecessary outages (the parent resource goes offline and takes the dependent resource down with it).

Figure 12 shows the default dependency tree for a SQL Server instance.



**Figure 12:** The default dependency tree for a SQL Server instance

### Evaluate Resource Failover Policies

Resource groups fail over as a single unit. In other words, if one resource fails, it has the potential to take everything else in that resource group down with it. For example, if your SQL Server resource fails to come online on one node of the cluster, you would want the SQL Server outage to force a failover of the entire resource group. However, if an MSDTC resource in your SQL Server resource group failed, and it was only being used for some noncritical remote queries, you would not want it failing over the entire resource group.

Evaluate each resource’s policy in the SQL Server resource group and decide whether the availability of this resource is critical enough to warrant a failover of the entire group.

#### Required Resources

The following resources are required in any SQL Server resource group:

* The SQL Server resource.
* Physical disk resources that store SQL Server data and log files (including disks that host mount points).
* An IP address that supports the SQL Server network name resource.
* A network name that hosts SQL Server.

Generally you will leave these resources at their default settings. You must set the policy on any additional physical disks that are added to your SQL Server instance after setup. Make sure that each resource policy is set to **If resource fails, attempt restart on current node** and select the **If restart is unsuccessful, fail over all resources in this service or application** check box.

#### Other Resources

These resources require further evaluation of your environment to determine whether their availability is critical enough to warrant a failover of the SQL Server resource group.

For maximum availability, the recommendation for optional resources is to set them to **If resource fails, attempt restart on current node** and clear the **If restart is unsuccessful, fail over all resources in this service or application** check box. This enables restarts for the resource, but prevents a failover event of the entire group that would take down the SQL Server instance:

* **SQL Server Agent**. If the SQL Server Agent resource fails, scheduled jobs such as backups, database maintenance, and replication jobs will not run. If SQL Server Agent is being used for such critical tasks, the recommendation is to allow this resource to affect the group to prevent missed backups and maintenance jobs.
* **File shares**. SQL Server uses file share resources for FileStream, log shipping, and replication. If the file share resource fails, these features may not function correctly.
* **Analysis Services**. Analysis Services is not dependent on the SQL Server resource.

### Connectivity Configuration

All SQL Server named instances use dynamic port allocation by default. Clients obtain this port information from the SQL Server Browser. The SQL Server Browser service is required for dynamic port allocation to work correctly.

If you assign static IP ports to all of your SQL instances, you will need to do one of two things:

1. Enable the SQL Server Browser service on each node of the cluster.
2. Create an alias on all clients, including all nodes of the cluster, which specifies the port number of the SQL Server instance. Client applications can alternatively add the port number in the connection string.

If you are using dynamic port allocation and the SQL Server Browser service is running on each node of the cluster, but you are still having connectivity problems, make sure that you have configured your network security to allow discovery traffic. The discovery process works as follows:

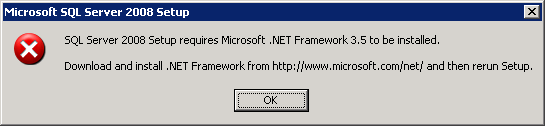
1. The SQL Server Browser listens on User Datagram Protocol (UDP) port 1434 across all IP addresses (IPAll).
2. The client sends a UDP packet to UDP port 1434 on the SQL Server IP address (the IP that corresponds to the SQL Server network name).
3. SQL Server Browser responds to the client IP address from the physical IP address of the node (not the SQL Server IP address). In other words, the client initiates a connection to the SQL Server IP address, but gets a response from the physical IP address of the node.
4. Depending on the security configuration of the client, server, and network, firewalls or IPSec may drop this response packet because the response IP address has changed.

For more information about troubleshooting SQL Server failover clusters, see [Failover Cluster Troubleshooting](http://technet.microsoft.com/en-us/library/ms189117.aspx) (http://technet.microsoft.com/en-us/library/ms189117.aspx) in SQL Server 2008 Books Online.

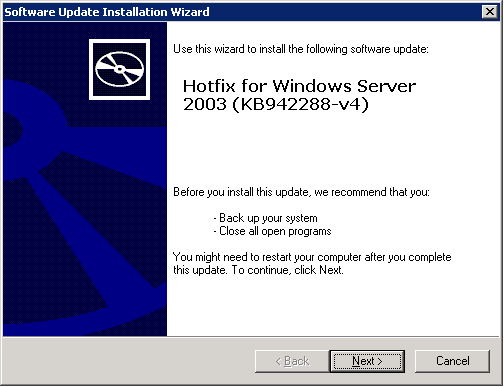
# Appendix A: Installing a SQL Server 2008 Failover Cluster Using Integrated Installation with Add Node Operation

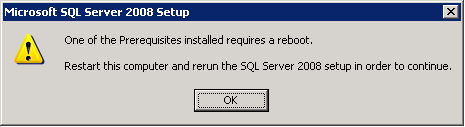
1. Insert the SQL Server installation media, and from the root folder, double-click Setup.exe. To install from a network share, browse to the root folder on the share, and then double-click Setup.exe. For more information about how to install prerequisites, see [Before Installing Failover Clustering](http://msdn.microsoft.com/en-us/library/ms189910.aspx) (http://msdn.microsoft.com/en-us/library/ms189910.aspx). You may be asked to install the prerequisites, if they are not previously installed.

.NET Framework 3.5 SP1:

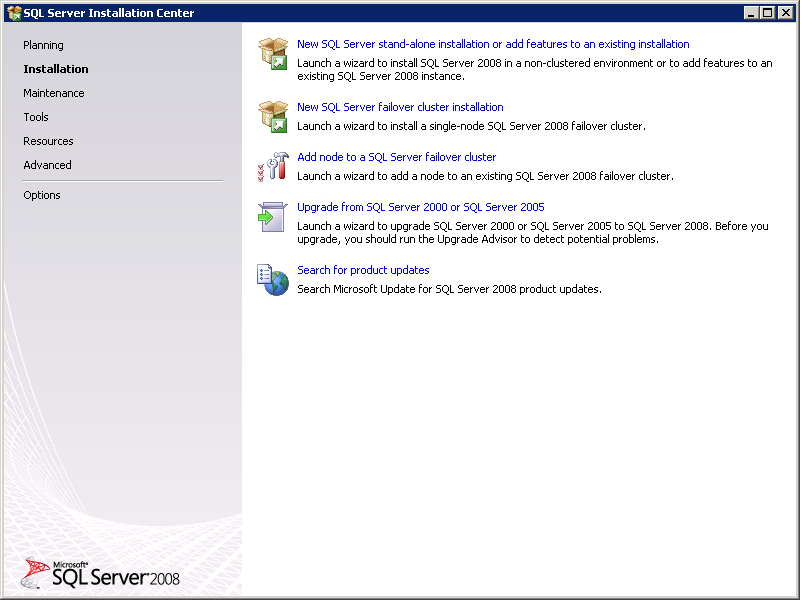


Install Windows Installer 4.5: Windows Installer 4.5 is required, and it can be installed by the Installation Wizard. If you are prompted to restart your computer, restart it, and then start SQL Server 2008 Setup again. If you want to preinstall the Windows Installer 4.5 package, you can download it from here: <http://support.microsoft.com/default.aspx/KB/942288>

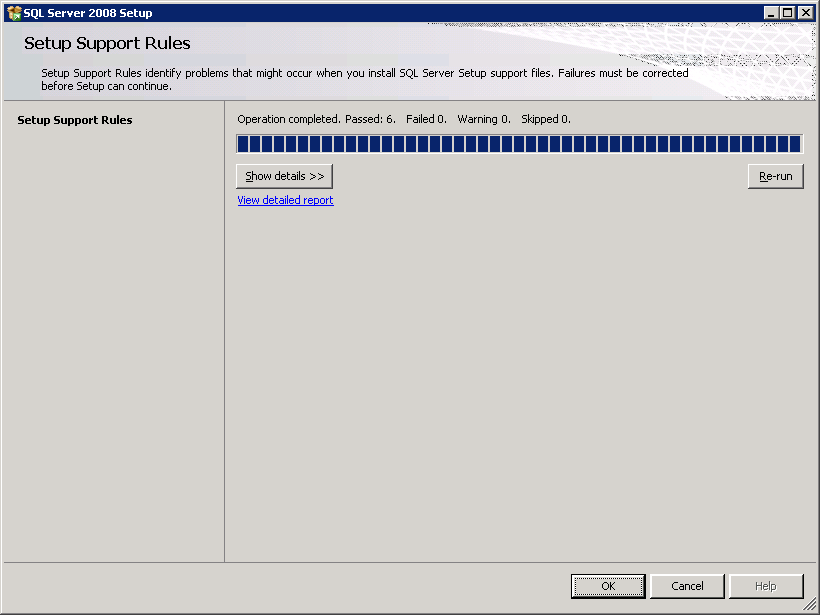


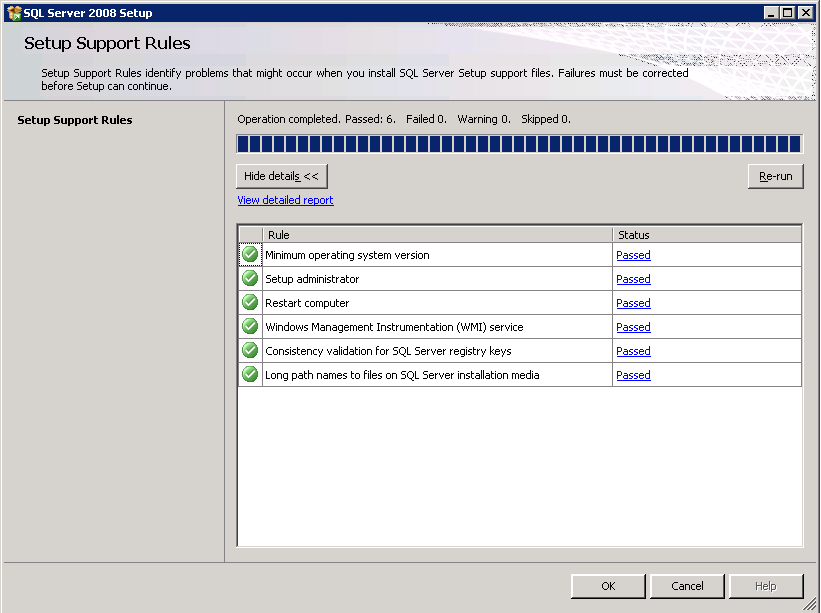


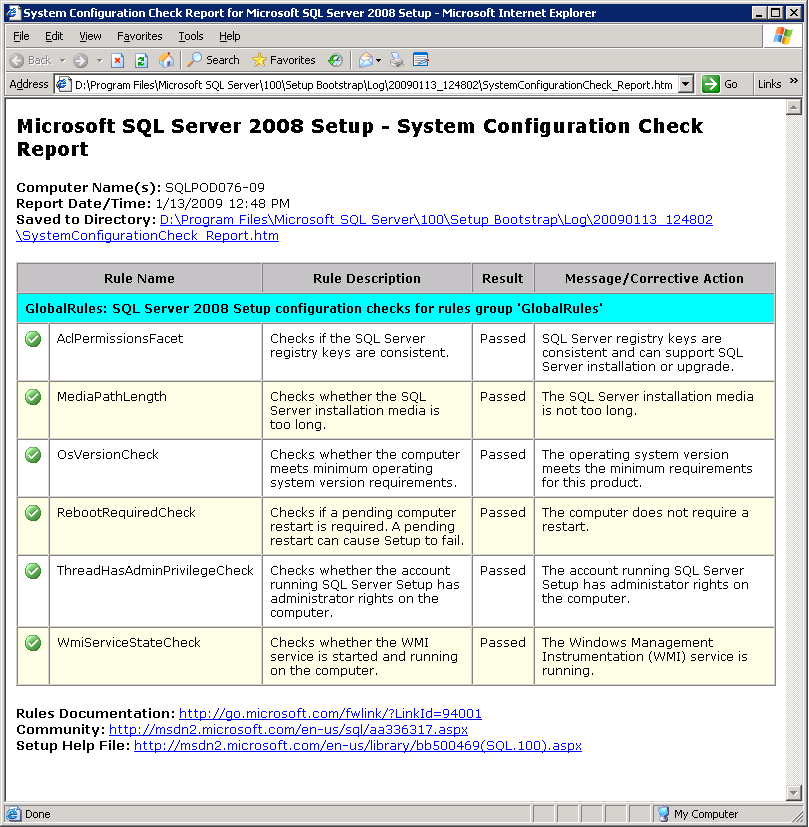
1. After prerequisites are installed, the Installation Wizard starts the SQL Server Installation Center. To create a new cluster installation of SQL Server 2008, click **New SQL Server failover cluster installation** on the Installation page.



1. The System Configuration Checker runs a discovery operation on your computer. To continue, click **OK**. You can view the details on the screen by clicking **Show Details**, or as an HTML report by clicking **View detailed report**.

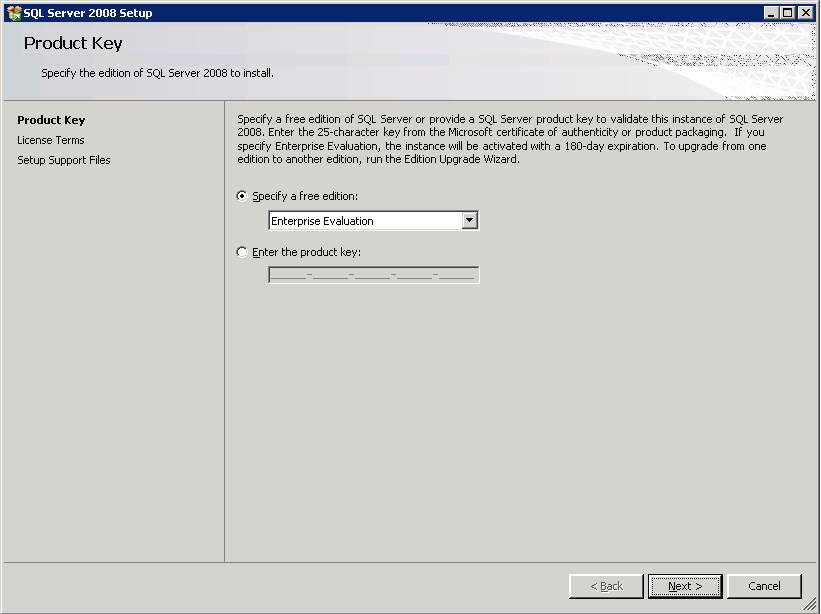




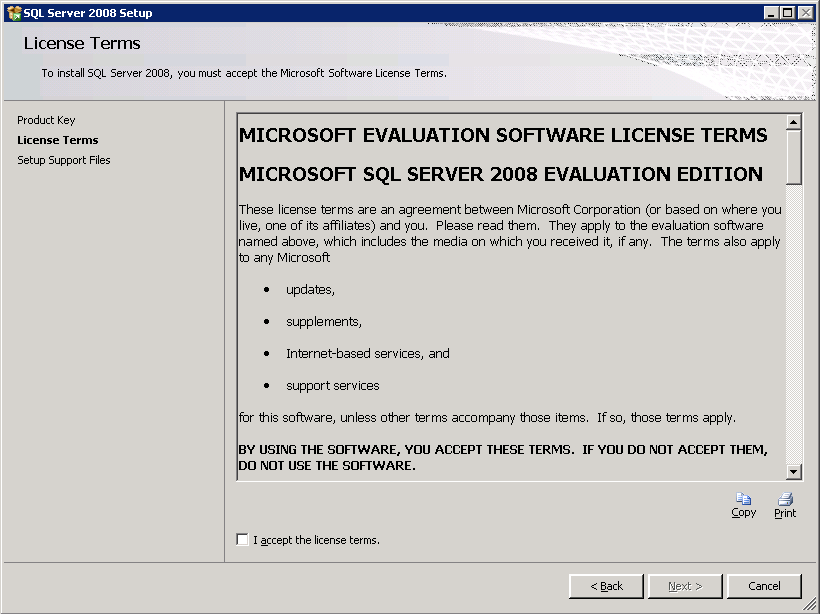


1. On the Product key page, indicate whether you are installing a free edition of SQL Server, or whether you have a PID key for a production version of the product. For more information, see [Editions and Components of SQL Server 2008](http://technet.microsoft.com/en-us/library/ms144275.aspx) (http://technet.microsoft.com/en-us/library/ms144275.aspx).

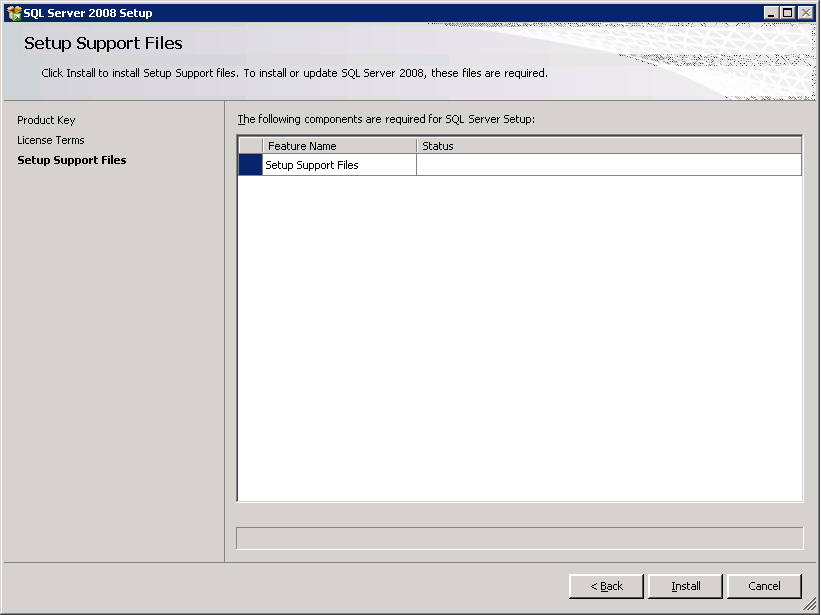
**Note**: The Product Key and License Terms pages show up after the setup support files page if you already installed the setup support files during a previous installation.



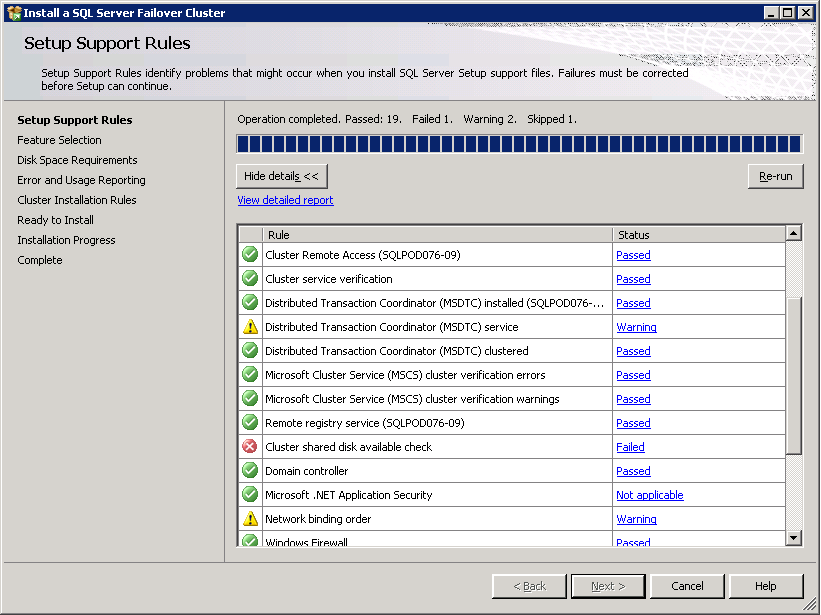
1. On the License Terms page, read the license agreement, and then select the check box to accept the license terms and conditions. Click **Next** to continue. To end Setup, click **Cancel**.



1. On the Setup Support Files page, click **Install** to install the setup support files.

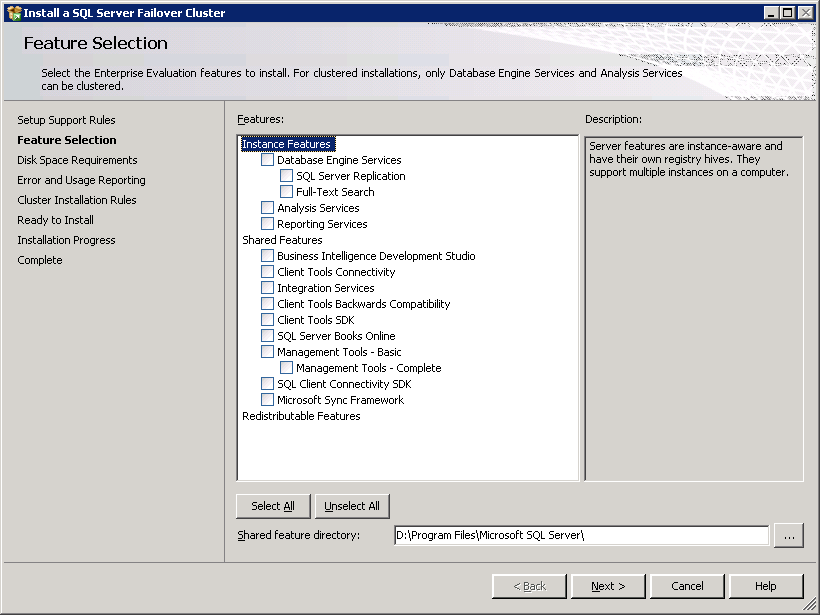


1. The System Configuration Checker verifies the system state of your computer before Setup continues. After the check is complete, click **Next** to continue. You can view the details on the screen by clicking **Show Details**, or as an HTML report by clicking **View detailed report**.



Correct any issues that are reported on the rules list. Errors block Setup, but warnings do not. It is a best practice to address all warnings and errors.

1. On the Feature Selection page, select the components for your installation.

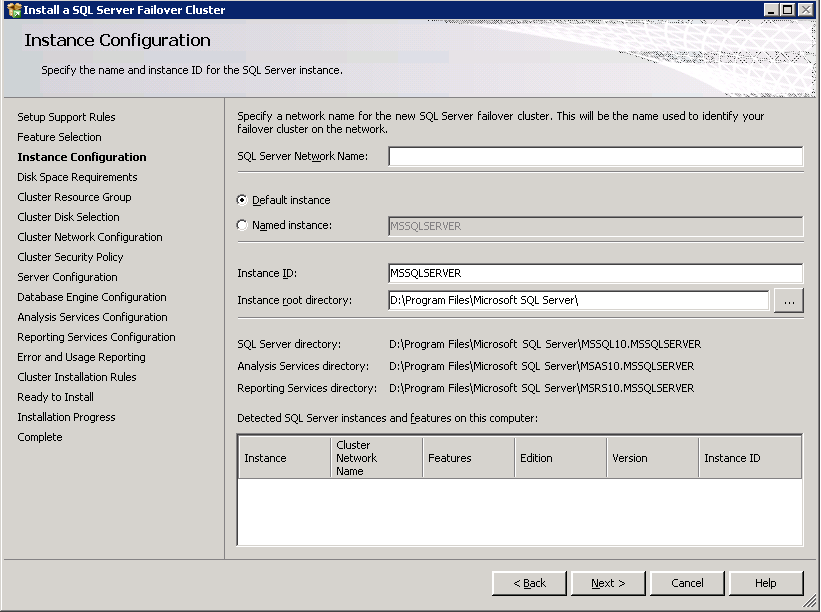


A description for each component group appears in the right pane after you select the feature name. You can select any combination of check boxes, but only the Database Engine and Analysis Services support failover clustering. Other selected components will run as stand-alone features without failover capability on the current node that you are running Setup on.

You can specify a custom directory for shared components by using the field at the bottom of this page. To change the installation path for shared components, either update the path in the field provided at the bottom of the dialog box, or click the ellipsis button to browse to an installation directory. The default installation path is C:\Program Files\Microsoft SQL Server\.

**Note:** When you select the Database Engine Services feature, both replication and full-text search are selected automatically. Unselecting any of these subfeatures also unselects the Database Engine Services feature.

1. On the Instance Configuration page, specify whether to install a default or a named instance.



**SQL Server Network Name** — Specify a network name for the new SQL Server failover cluster. This is the name that is used to identify your failover cluster on the network.

**Note:** This is known as the virtual SQL Server name in earlier versions of SQL Server failover clusters.

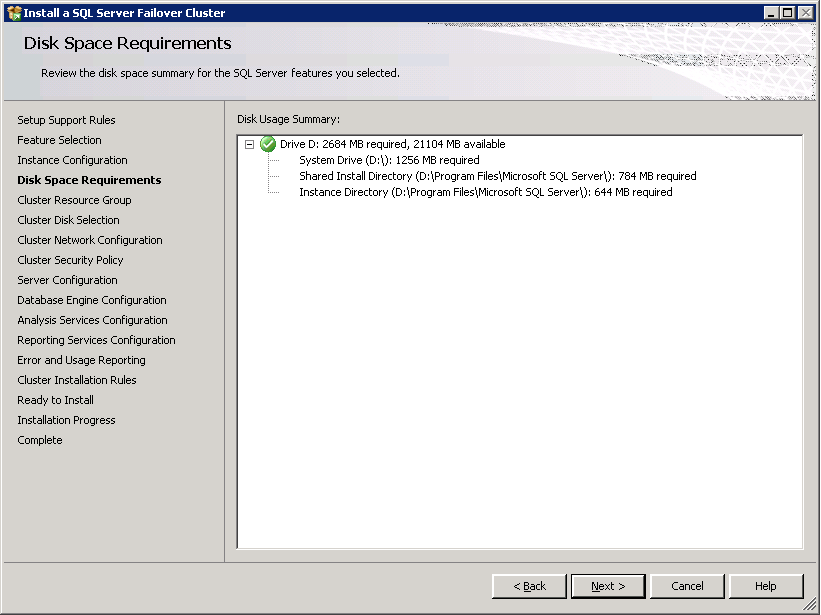
**Instance ID** — By default, the instance name is used as the Instance ID. This is used to identify installation directories and registry keys for your instance of SQL Server. This is the case for default instances and named instances. For a default instance, the instance name and instance ID would be MSSQLSERVER. To use a non-default instance ID, select the **Instance ID** box and provide a value.

**Note:** Typical stand-alone instances of SQL Server 2008, whether default or named instances, do not use a nondefault value for the **Instance ID** box.

**Instance root directory** — By default, the instance root directory is C:\Program Files\Microsoft SQL Server\. To specify a nondefault root directory, use the field provided, or click the ellipsis button to locate an installation folder.

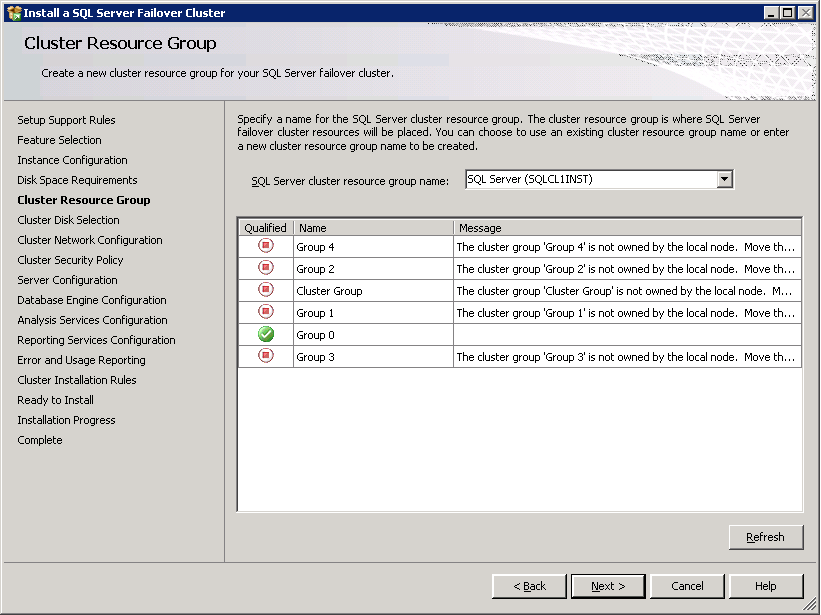
**Detected SQL Server instances and features on this computer** - The grid shows instances of SQL Server that are on the computer where Setup is running. If a default instance is already installed on the computer, you must install a named instance of SQL Server 2008. Click **Next** to continue.

1. The Disk Space Requirements page calculates the required disk space for the features that you specify, and it compares requirements to the available disk space on the computer where Setup is running.

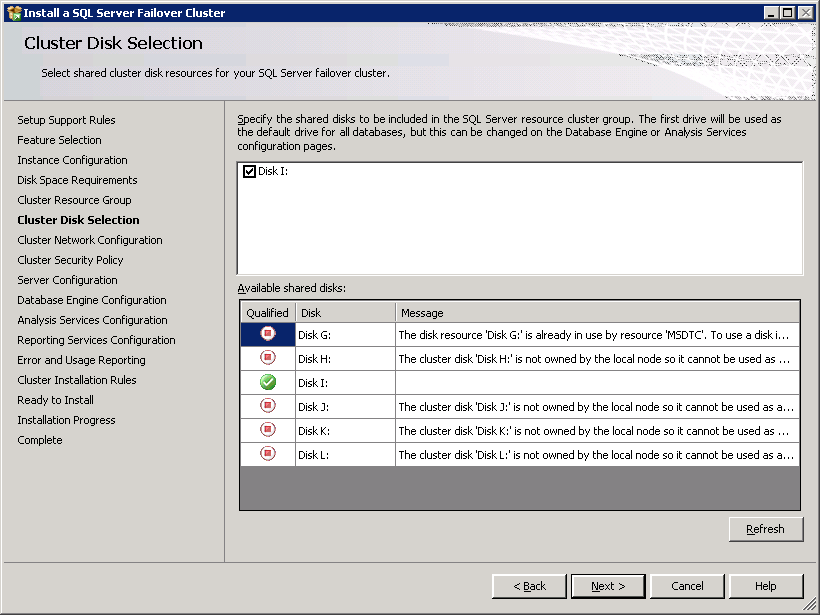


1. Use the Cluster Resource Group page to specify the cluster resource group name where SQL Server virtual server resources will be located. To specify the SQL Server cluster resource group name, you have two options:

* Use the drop-down box to specify an existing group to use.
* Type the name of a new group to create.



1. On the Cluster Disk Selection page, select the shared cluster disk resource for your SQL Server failover cluster.



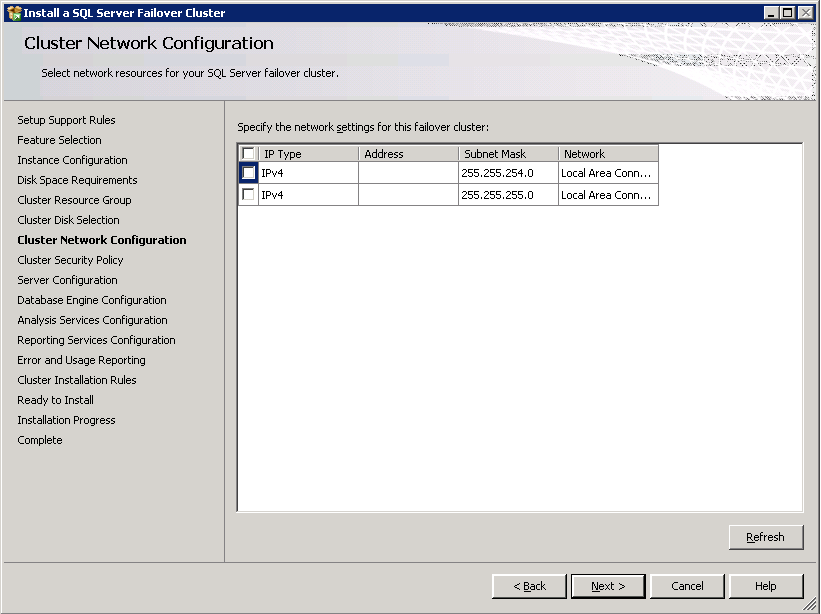
The cluster disk is where the SQL Server data will be stored. More than one disk can be specified. The **Available shared disks** box displays a list of available disks, whether each is qualified as a shared disk, and a description of each disk resource. Click **Next** to continue.

**Note:** The first drive is used as the default drive for all databases, but it can be changed on the Database Engine or Analysis Services configuration pages.

1. On the Cluster Network Configuration page, specify the network resources for your failover cluster instance:

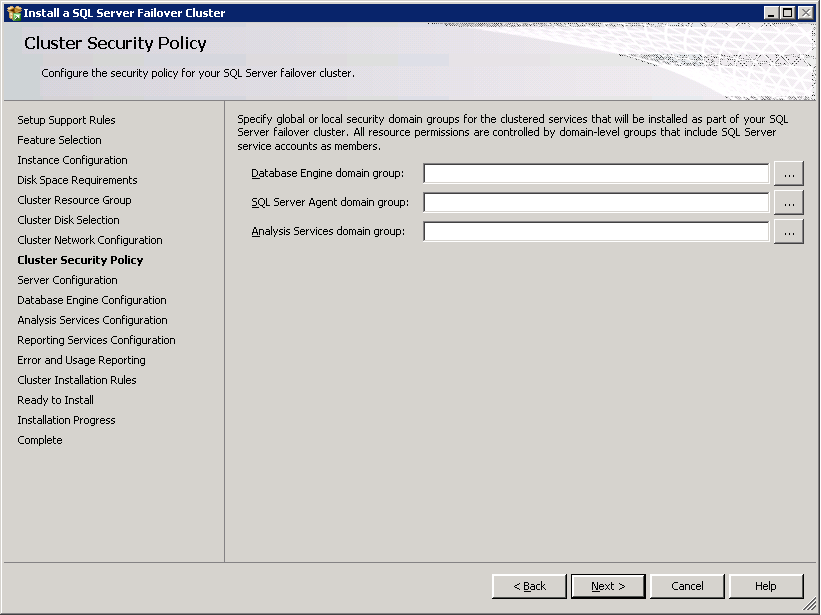
* **Network Settings** — Specify the IP type and IP address for your failover cluster instance. On Windows Server 2008 failover clusters, SQL Server supports the use of DHCP addresses. Before choosing this option make sure that any network security such as firewalls and IPsec in use between this SQL Server and its clients can accommodate server side DHCP. For example, some firewalls will require a static port for proper configuration of the firewall.

Click **Next** to continue.

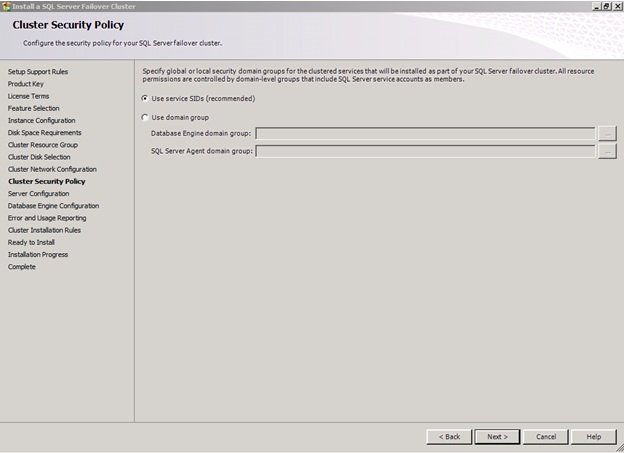


1. Use the following page to specify the security policy for the cluster.

The following screenshot displays the cluster security policies for Windows Server 2003. In Windows Server 2003, you can not leverage service SIDs. Specify domain groups for SQL Server services. All resource permissions are controlled by domain-level groups that include SQL Server service accounts as group members. This is displayed in the following screen shot.



The following screenshot displays the cluster security policies available for Windows Server 2008. In Windows Server 2008 and later versions, service SIDs (server security IDs) are the recommended and default setting. The option to specify domain groups is available but not recommended. For information about service SIDs functionality on Windows Server 2008, see [Setting Up Windows Service Accounts](http://msdn.microsoft.com/en-us/library/ms143504.aspx) (http://msdn.microsoft.com/en-us/library/ms143504.aspx). This is displayed in the following screen shot.



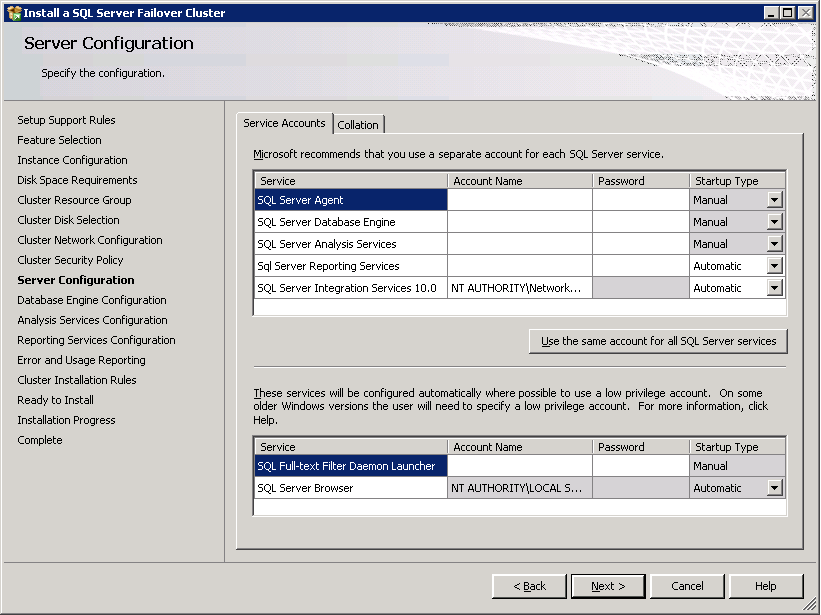
Click **Next** to continue.

**Note:** If you are installing a SQL Server 2008 failover cluster instance in a Windows 2000 Server mixed-mode domain, you must use domain global groups for SQL Server clustered services.

Note: Windows 2000 Server domain controllers can operate in mixed mode or native mode. Mixed mode enables down-level domain controllers in the same domain.

Work flow for the rest of this procedure depends on the features that you have specified for your installation. You might not see all the pages, depending on your selections (Database Engine, Analysis Services, or Reporting Services).

1. On the **Service Accounts** tab, specify login accounts for SQL Server services. The actual services that are configured on this page depend on the features that you are installing.

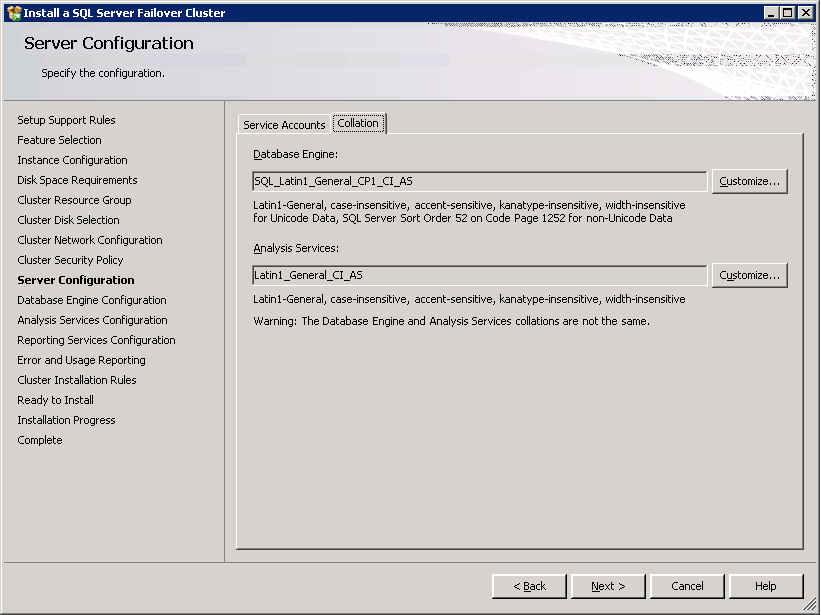


You can assign the same login account to all SQL Server services, or you can configure each service account individually. The startup type is set to manual for all cluster-aware services, including full-text search and SQL Server Agent, and cannot be changed during installation. Microsoft recommends that you configure service accounts individually to provide least privileges for each service, where SQL Server services are granted the minimum permissions they have to have complete their tasks. For more information, see [Setting Up Windows Service Accounts](http://msdn.microsoft.com/en-us/library/ms143504.aspx) (http://msdn.microsoft.com/en-us/library/ms143504.aspx)and [SQL Server Configuration - Service Accounts](http://msdn.microsoft.com/en-us/library/cc281953.aspx) (http://msdn.microsoft.com/en-us/library/cc281953.aspx)in SQL Server Books Online.

1. To specify the same login account for all service accounts in this instance of SQL Server, provide credentials in the fields at the bottom of the page.

**Security note:** Do not use a blank password. Use a strong password.

1. When you are finished specifying login information for SQL Server services, click **Next**.
2. Use the **Collation** tab to specify nondefault collations for the Database Engine and Analysis Services.

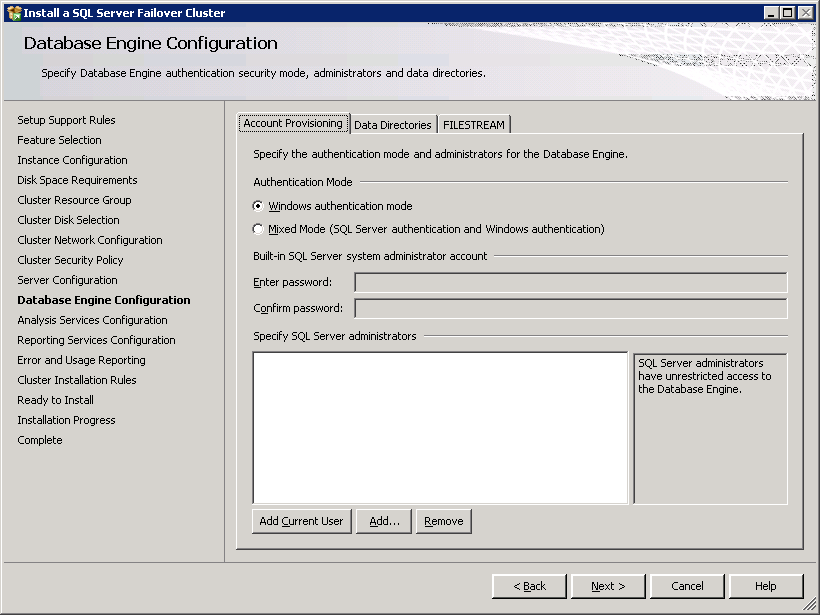


1. Use the **Account Provisioning** tab to specify the following:

* Authentication mode - select Windows authentication or Mixed Mode authentication for your instance of SQL Server. If you select Mixed Mode authentication, you must provide a strong password for the built-in SQL Server system administrator account.

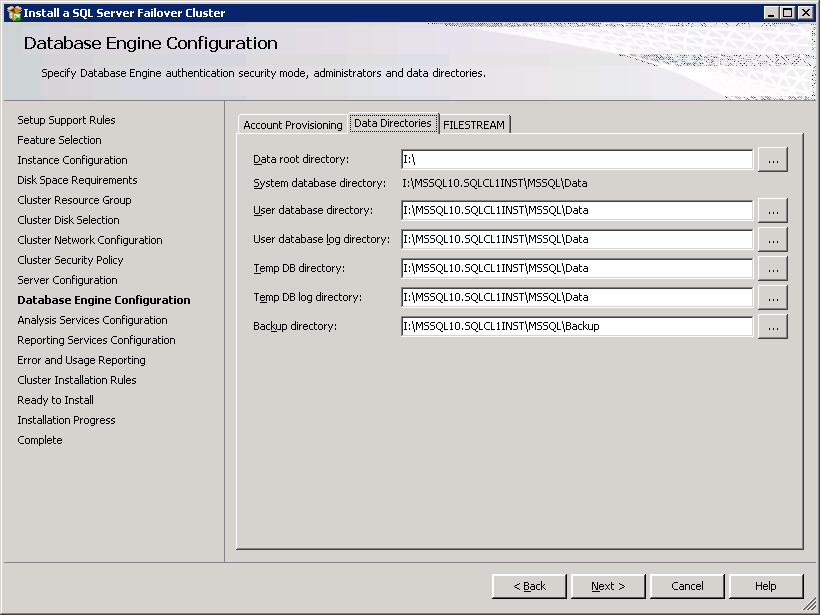
After a device establishes a successful connection to SQL Server, the security mechanism is the same for both Windows authentication and Mixed Mode. For more information, see [Database Engine Configuration - Account Provisioning](http://technet.microsoft.com/en-us/library/cc281849.aspx) (http://technet.microsoft.com/en-us/library/cc281849.aspx).

* SQL Server administrators - You must specify at least one system administrator for the instance of SQL Server. To add the account under which SQL Server Setup is running, click **Add Current User**. To add or remove accounts from the list of system administrators, click **Add** or **Remove**, and then edit the list of users, groups, or computers that will have administrator privileges for the instance of SQL Server. For more information, see [Database Engine Configuration - Account Provisioning](http://technet.microsoft.com/en-us/library/cc281849.aspx) (http://technet.microsoft.com/en-us/library/cc281849.aspx).



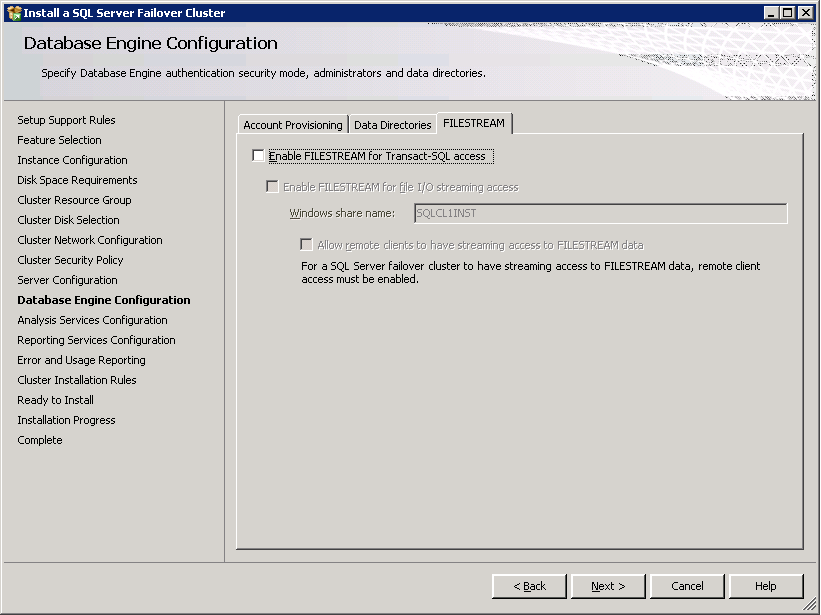
When you are finished editing the list, click **OK**. Verify the list of administrators in the configuration dialog box. When the list is complete, click **Next**.

1. Use the **Data Directories** tab to specify nondefault installation directories. To install to default directories, click **Next**.



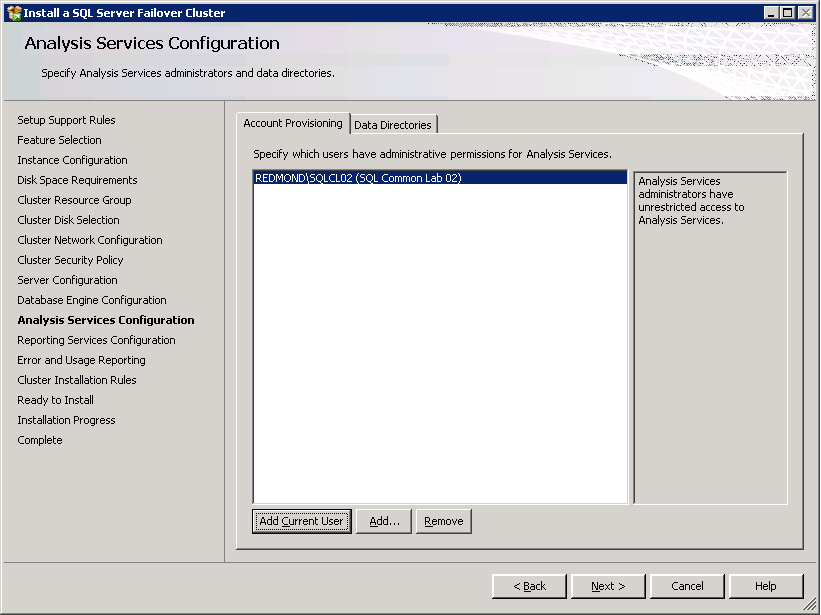
**Important:** If you specify nondefault installation directories, make sure that the installation folders are unique to this instance of SQL Server. None of the directories in this dialog box should be shared with directories from other instances of SQL Server. The data directories should be located on the shared cluster disk for the failover cluster.

1. Use the **FILESTREAM** tab to enable FILESTREAM for your instance of SQL Server.



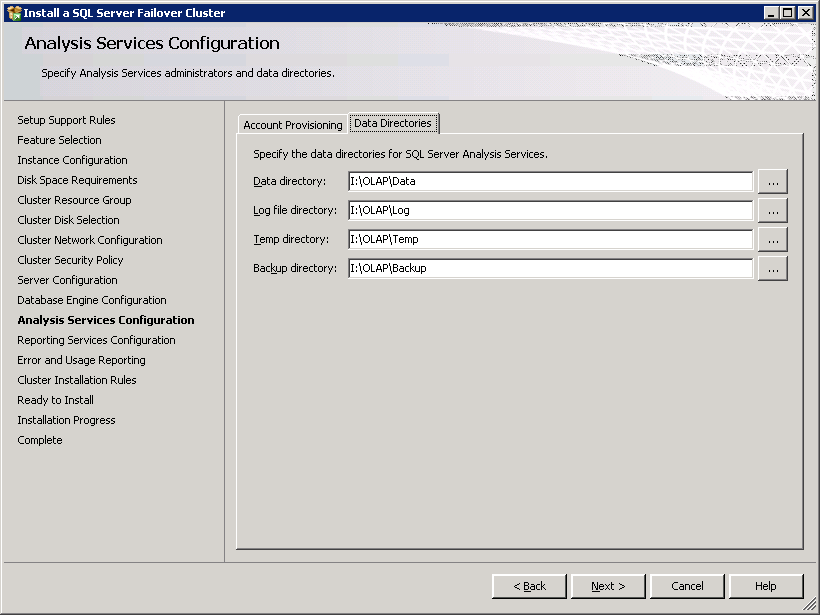
Click **Next** to continue.

1. On the Analysis Services Configuration page, use the **Account** **Provisioning** tab to specify users or accounts that will have administrator permissions for Analysis Services. You must specify at least one system administrator for Analysis Services. To add the account under which SQL Server Setup is running, click **Add Current User**. To add or remove accounts from the list of system administrators, click **Add** or **Remove**, and then edit the list of users, groups, or computers that will have administrator privileges for Analysis Services.



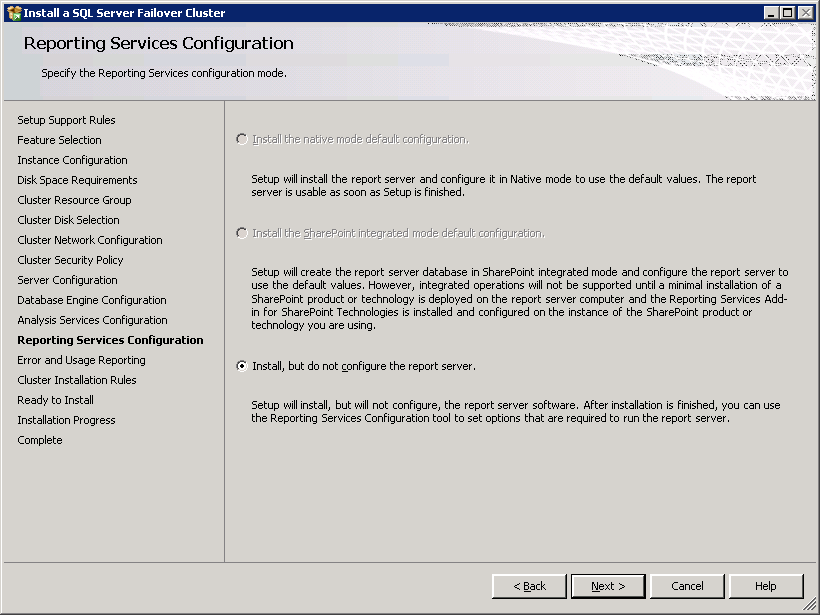
When you are finished editing the list, click **OK**. Verify the list of administrators in the configuration dialog box. When the list is complete, click **Next**.

1. Use the **Data Directories** tab to specify nondefault installation directories. To install to default directories, click **Next**.

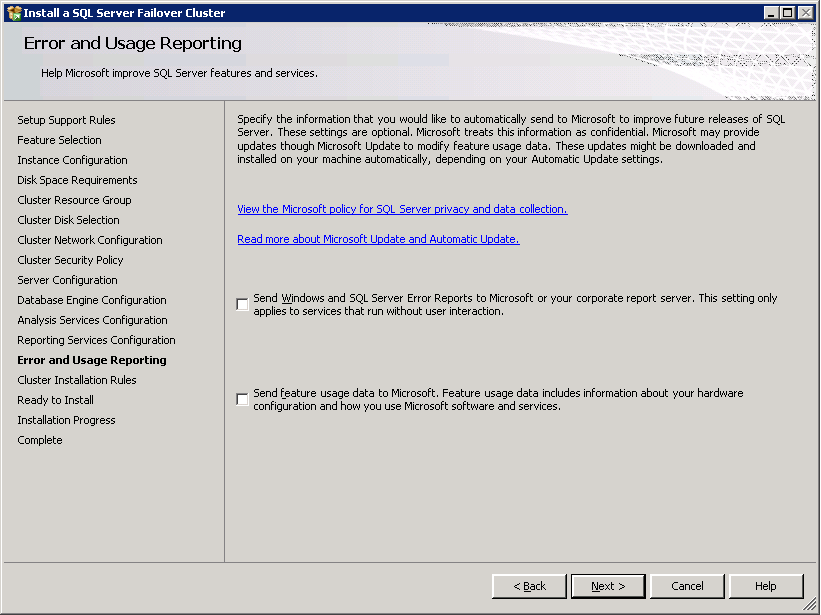


**Important:** If you specify nondefault installation directories, make sure that the installation folders are unique to this instance of SQL Server. None of the directories in this dialog box should be shared with directories from other instances of SQL Server. The data directories should be located on the shared cluster disk for the failover cluster.

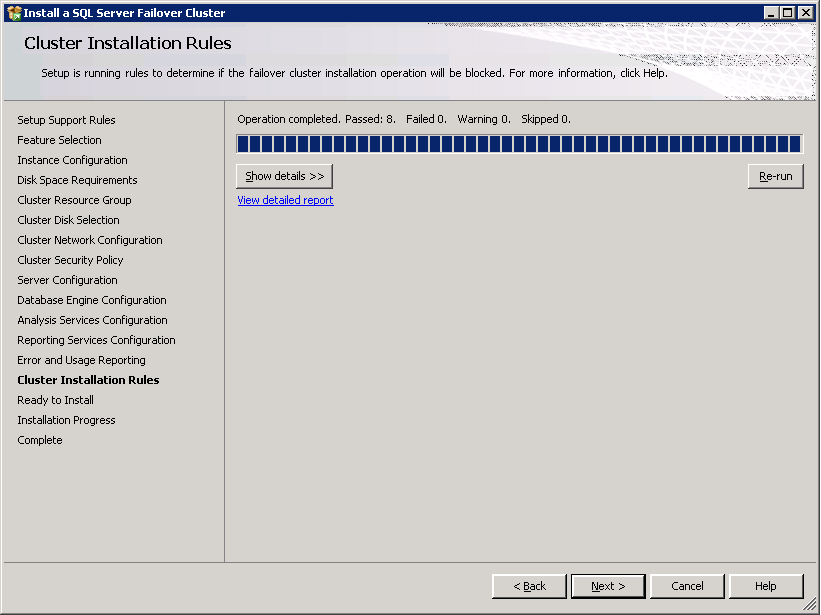
1. Use the Reporting Services Configuration page to specify the kind of Reporting Services installation to create. For failover cluster installation, the option is set to “Install, but do not configure the report server.” You must configure Reporting Services after you complete the installation.



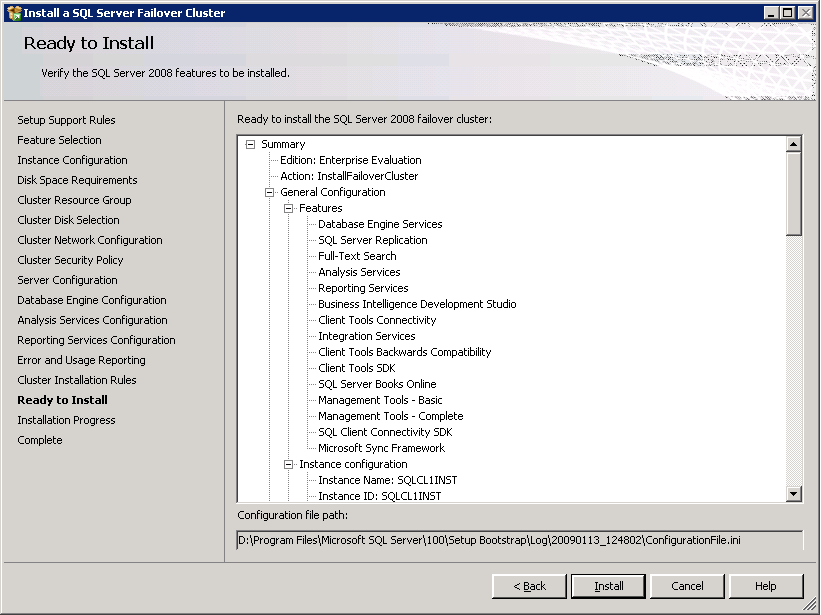
1. On the Error and Usage Reporting page, specify the information that you want to send to Microsoft that will help improve SQL Server. By default, options for error reporting and feature usage are disabled.



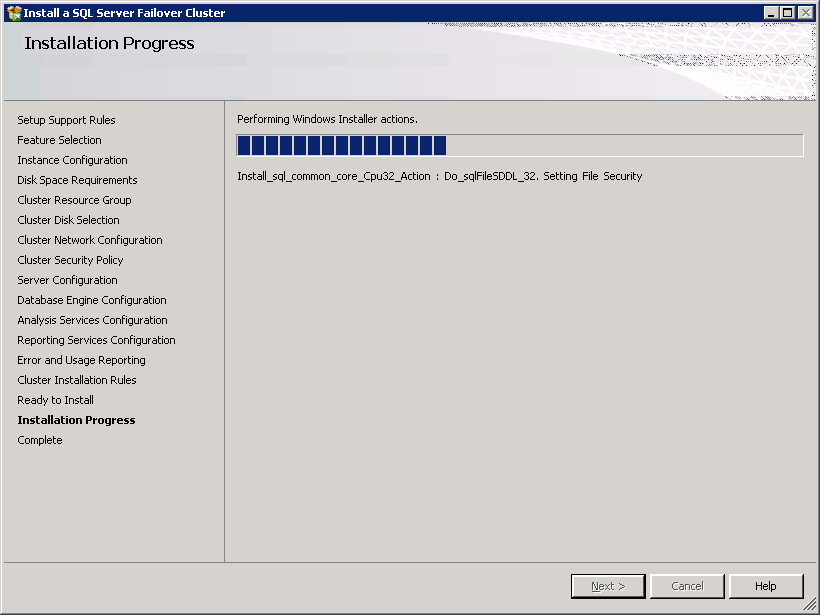
1. The System Configuration Checker runs one more set of rules to validate your configuration with the SQL Server features that you have specified.



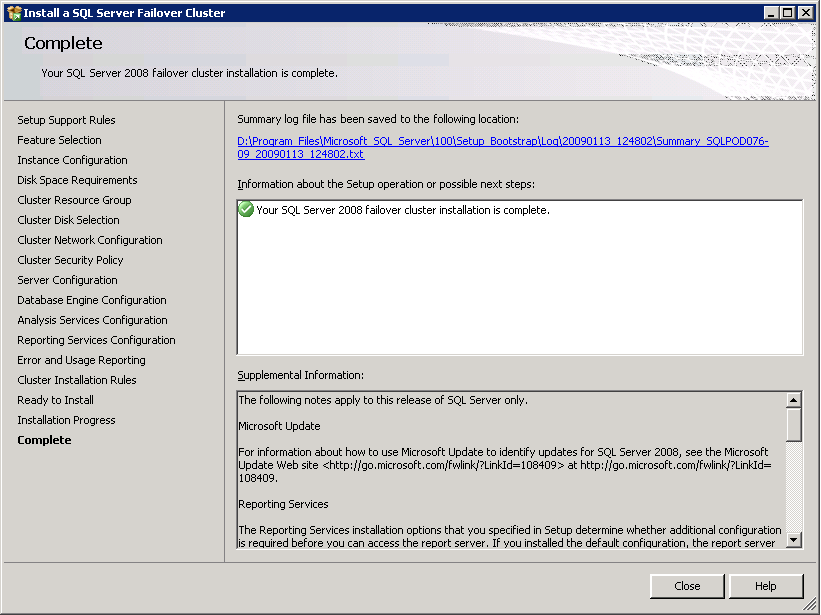
1. The Ready to Install page displays a tree view of installation options that you specified during Setup. To continue, click **Install**.



1. During installation, the Installation Progress page provides status so that you can monitor installation progress as Setup continues.



1. After installation, the Complete page provides a link to the summary log file for the installation and other important notes. To complete the SQL Server installation process, click **Close**.



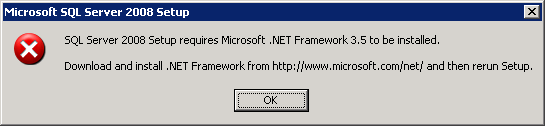
1. If you are instructed to restart the computer, do so now. It is important to read the message from the Installation Wizard when you have finished with Setup. For more information about setup log files, see [How To: View SQL Server Setup Log Files](http://msdn.microsoft.com/en-us/library/ms143702.aspx) (http://msdn.microsoft.com/en-us/library/ms143702.aspx).
2. To add nodes to the single-node failover you just created, run Setup on each additional node and follow the steps for Appendix C: Add Node. For more information, see [How to: Add or Remove Nodes in a SQL Server Failover Cluster (Setup*)*](http://msdn.microsoft.com/en-us/library/ms191545.aspx) (http://msdn.microsoft.com/en-us/library/ms191545.aspx)*.*

**Note:** The SQL Server edition you are installing must match across all the nodes in a SQL Server failover cluster. When you add a new node to an existing SQL Server failover cluster, make sure that the edition matches the edition of the existing failover cluster.

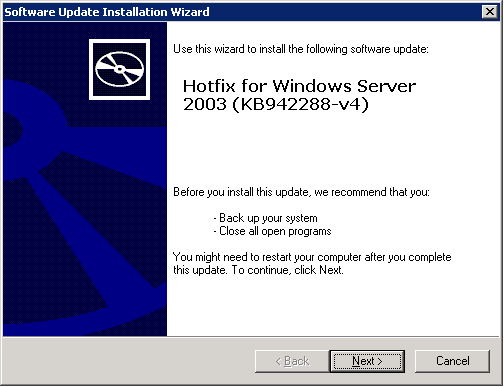
# Appendix B: Installing a SQL Server 2008 Failover Cluster Using Advanced/Enterprise Installation

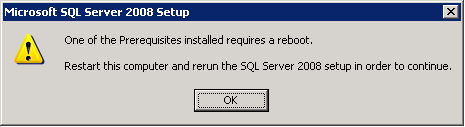
## Advanced/Enterprise Failover Cluster Install Step 1: Prepare

1. Insert the SQL Server installation media, and from the root folder, double-click Setup.exe. To install from a network share, browse to the root folder on the share, and then double-click Setup.exe. For more information about how to install prerequisites, see [Before Installing Failover Clustering](http://msdn.microsoft.com/en-us/library/ms189910.aspx) (http://msdn.microsoft.com/en-us/library/ms189910.aspx). You may be asked to install the prerequisites, if they are not previously installed.



Install Windows Installer 4.5. Windows Installer 4.5 is required, and it can be installed by the Installation Wizard. If you are prompted to restart your computer, restart it, and then start SQL Server 2008 Setup again.

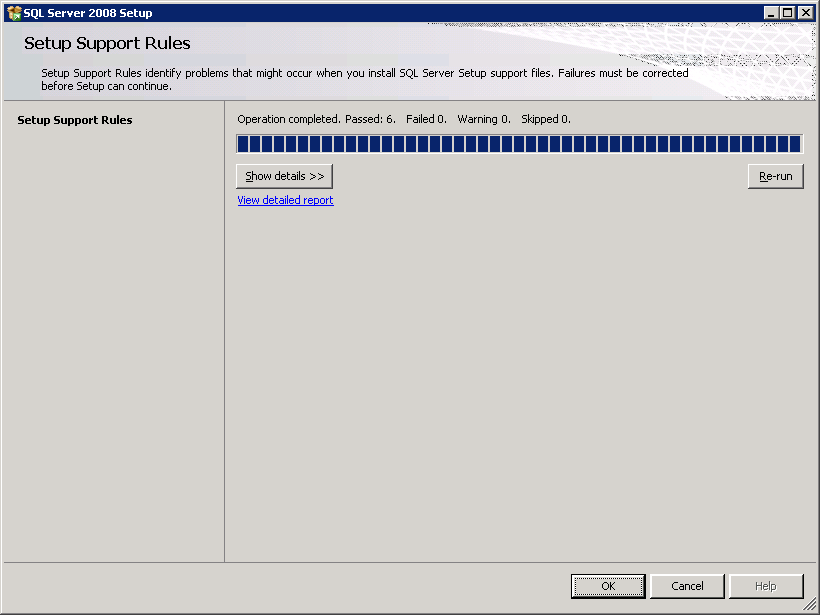


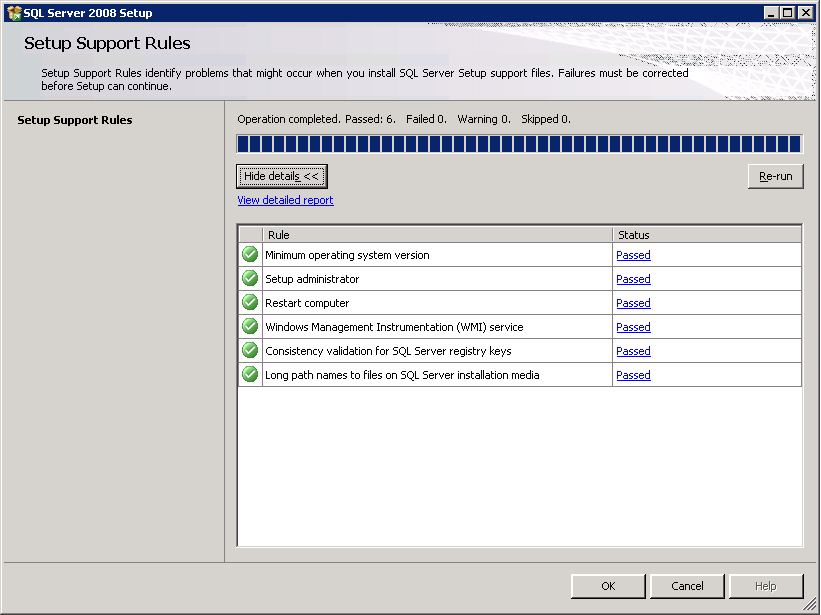


1. After the prerequisites are installed, the Installation Wizard starts the SQL Server Installation Center. To prepare the node for clustering, move to the Advanced page and then click **Advanced cluster preparation**.

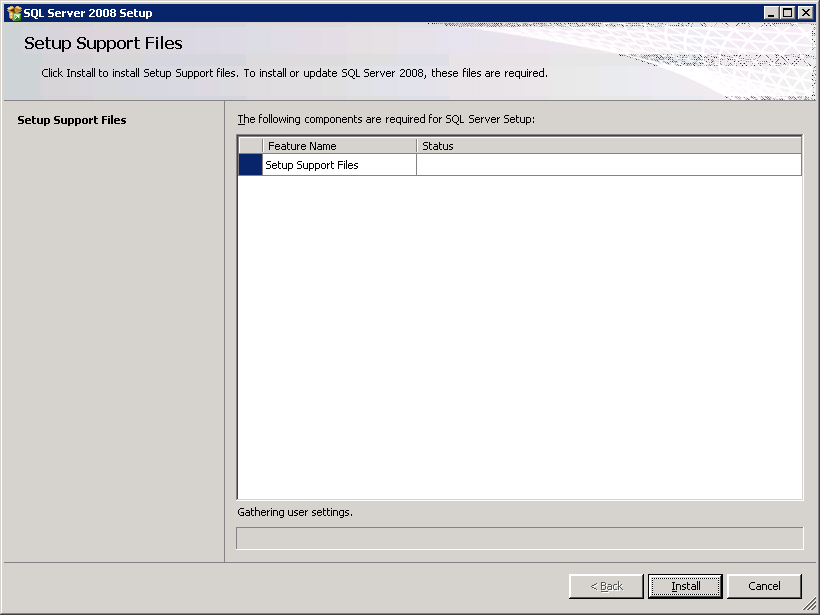


1. The System Configuration Checker runs a discovery operation on your computer. To continue, click **OK**. You can view the details on the screen by clicking **Show Details**, or as an HTML report by clicking **View detailed report**.

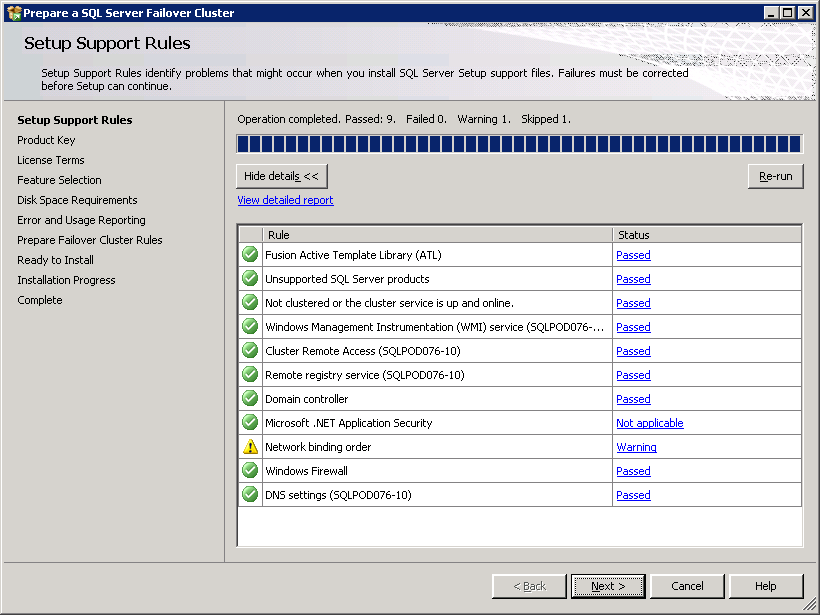




1. On the Setup Support Files page, click **Install** to install the setup support files.

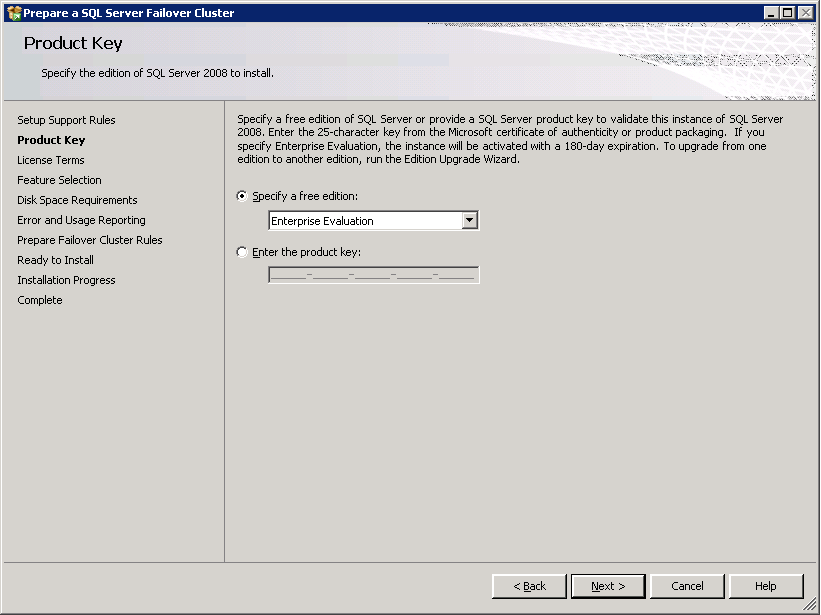


1. The System Configuration Checker verifies the system state of your computer before Setup continues. After the check is complete, click **Next** to continue. You can view the details on the screen by clicking **Show Details**, or as an HTML report by clicking **View detailed report**.



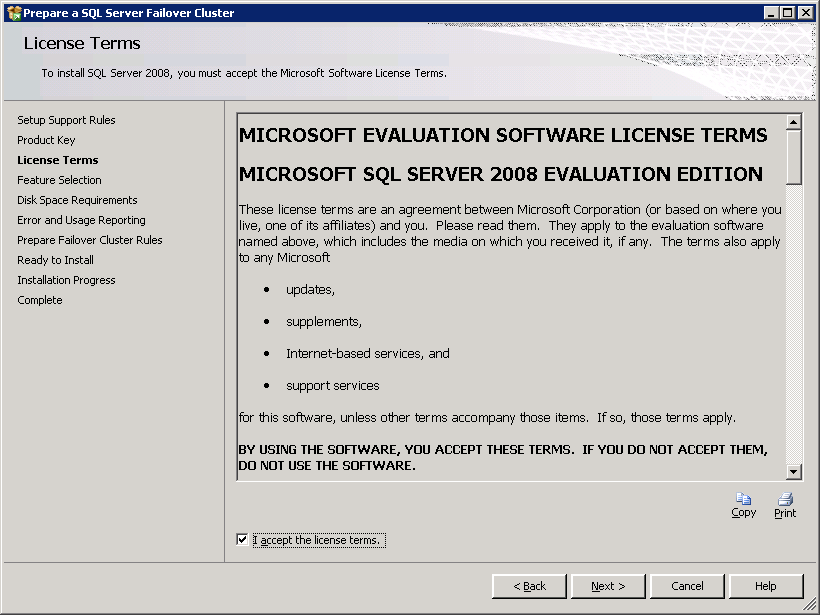
1. On the Product Key page, indicate whether you are installing a free edition of SQL Server or you have a PID key for a production version of the product.

**Note**: The Product Key and License Terms pages show up after the Setup Support Files page if you already installed the setup support files during a previous installation.



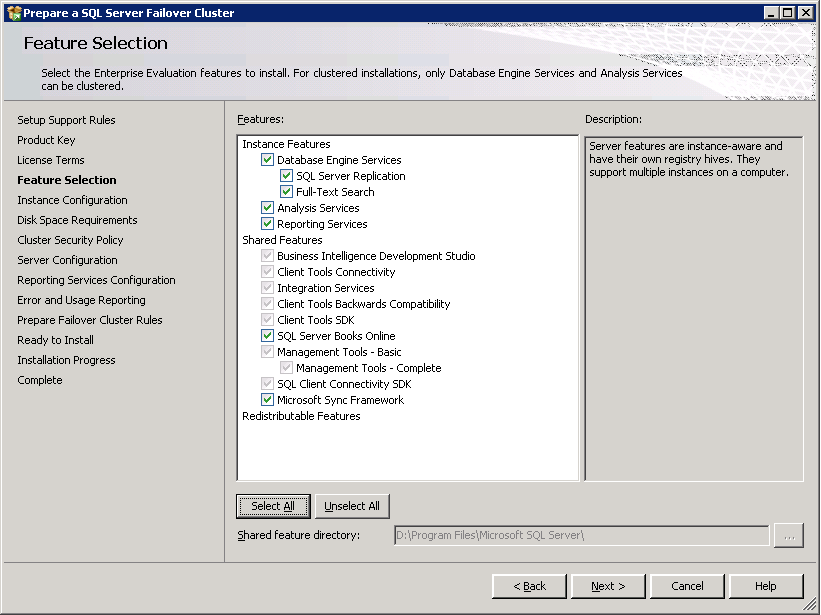
**Note:** You must specify the same product key on all the nodes that you are preparing for the same failover cluster.

1. On the License Terms page, read the license agreement, and then select the check box to accept the license terms and conditions.



Click **Next** to continue. To end Setup, click **Cancel**.

1. On the Feature Selection page, select the components for your installation.

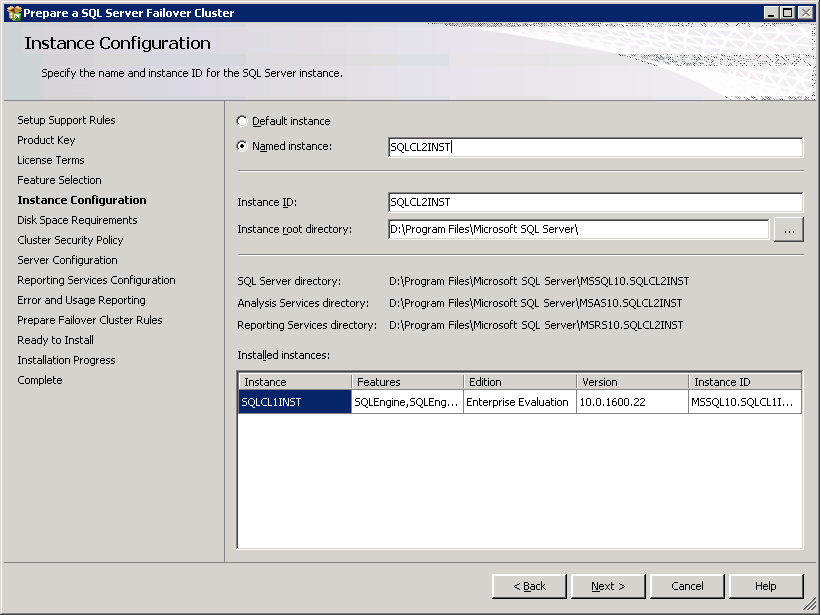


A description for each component group appears in the right pane after you select the feature name. You can select any combination of check boxes, but only the Database Engine and Analysis Services support failover clustering. Other components will run on a single failover cluster node as a stand-alone feature without failover capability.

You can specify a custom directory for shared components by using the field at the bottom of this page. To change the installation path for shared components, either update the path in the field provided at the bottom of the dialog box, or click the ellipsis button to browse to an installation directory. The default installation path is C:\Program Files\Microsoft SQL Server\.

**Note:** When you select the Database Engine Services feature, both replication and full-text search are selected automatically. Unselecting any of these subfeatures also unselects the Database Engine Services feature.

1. On the Instance Configuration page, specify whether to install a default or a named instance.



**Instance ID** — By default, the instance name is used as the Instance ID. This is used to identify installation directories and registry keys for your instance of SQL Server. This is the case for default instances and named instances. For a default instance, the instance name and instance ID would be MSSQLSERVER. To use a non-default instance ID, select the **Instance ID** text box and provide a value.

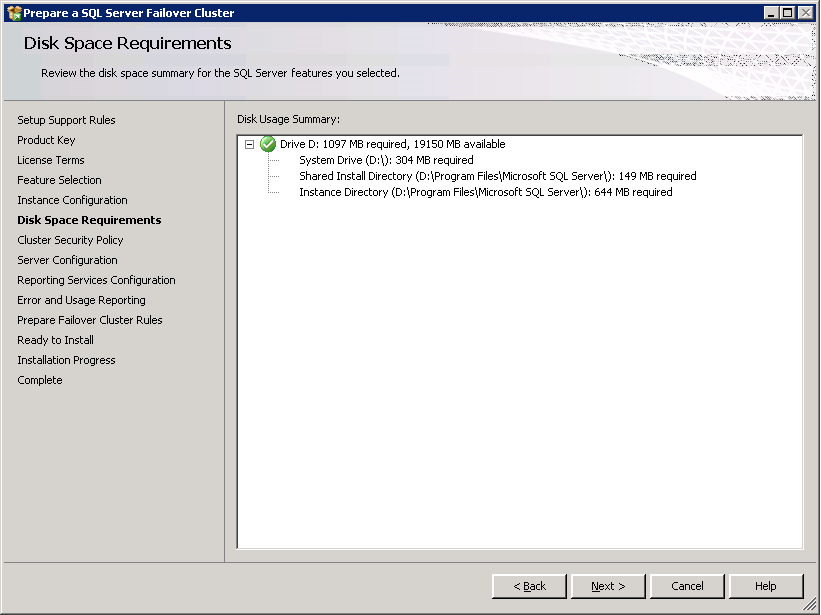
**Note:** Typical stand-alone instances of SQL Server 2008, whether default or named instances, do not use a non-default value for the **Instance ID** text box.

**Important:** Use the same InstanceID for all the nodes that are prepared for the failover cluster.

**Instance root directory** — By default, the instance root directory is C:\Program Files\Microsoft SQL Server\. To specify a nondefault root directory, use the field provided, or click the ellipsis button to locate an installation folder.

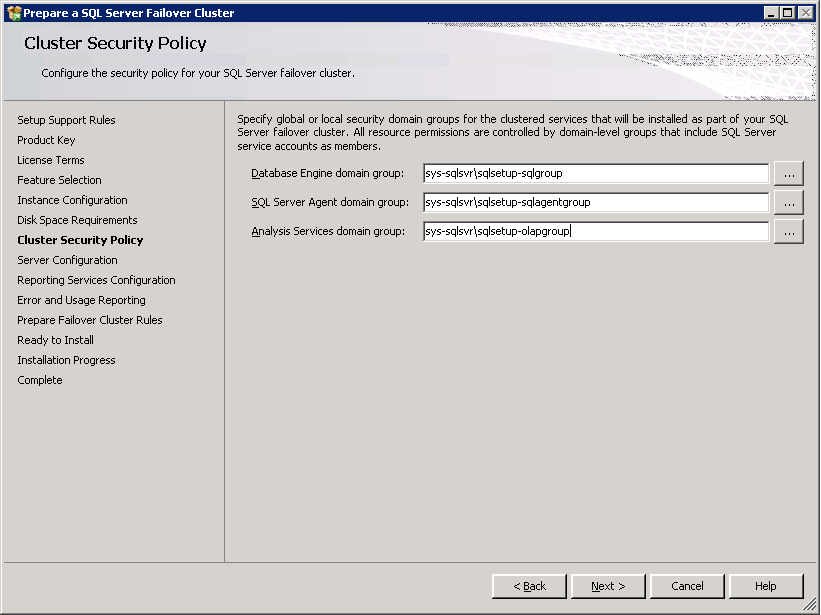
**Installed instances** - The box shows instances of SQL Server that are on the computer where setup is running. If a default instance is already installed on the computer, you must install a named instance of SQL Server 2008. Click **Next** to continue.

1. The Disk Space Requirements page calculates the required disk space for the features that you specify, and it compares requirements to the available disk space on the computer where Setup is running.

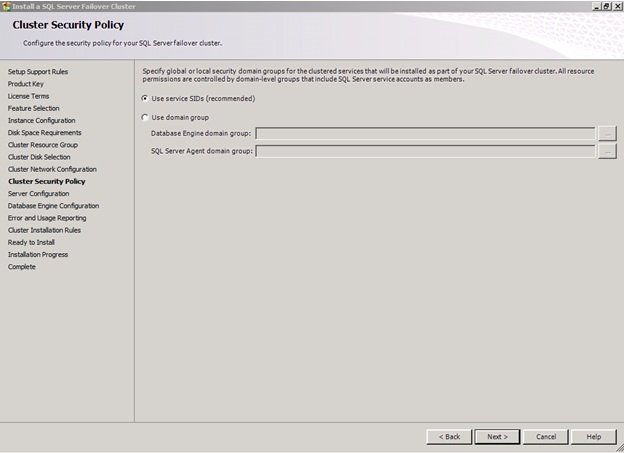


1. Specify the security policy for the cluster.

The following screenshot displays the cluster security policies for Windows Server 2003. In Windows Server 2003, you cannot leverage service SIDs. Specify domain groups for SQL Server services. All resource permissions are controlled by domain-level groups that include SQL Server service accounts as group members. This is displayed in the following screen shot.



The following screenshot displays the cluster security policies available for Windows Server 2008. In Windows Server 2008 and later versions, service SIDs (server security IDs) are the recommended and default setting. The option to specify domain groups is available but not recommended. For information about service SIDs functionality on Windows Server 2008, see [Setting Up Windows Service Accounts](http://msdn.microsoft.com/en-us/library/ms143504.aspx) (http://msdn.microsoft.com/en-us/library/ms143504.aspx). This is displayed in the following screen shot.



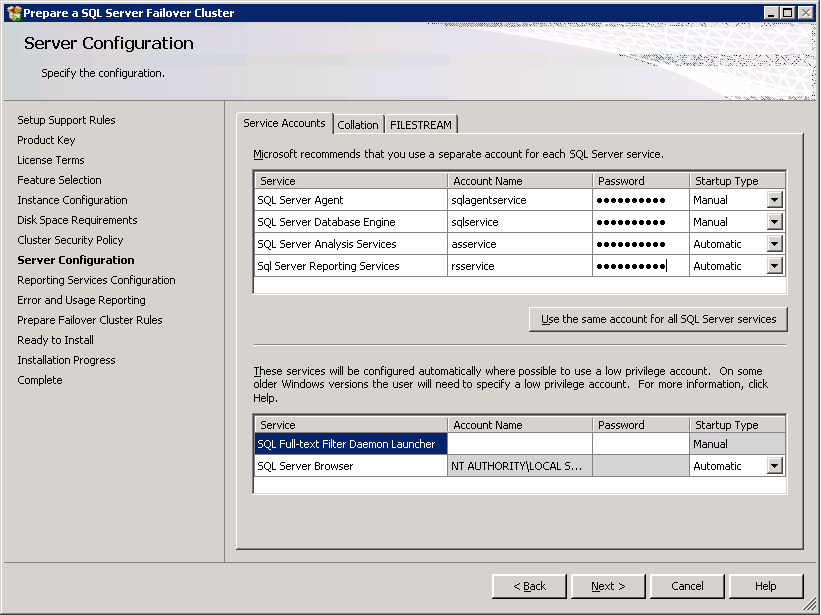
Click **Next** to continue.

**Note:** If you are installing a SQL Server 2008 failover cluster instance in a Windows 2000 Server mixed-mode domain, you must use domain global groups for SQL Server Clustered Services.

**Note**: Windows 2000 Server domain controllers can operate in mixed mode and native mode. Mixed mode enables down-level domain controllers in the same domain.

Work flow for the rest of this procedure depends on the features that you have specified for your installation. You might not see all the pages, depending on your selections.

1. On the Server Configuration page, on the **Service Accounts** tab, specify login accounts for SQL Server services. The actual services that are configured on this page depend on the features that you selected to install.



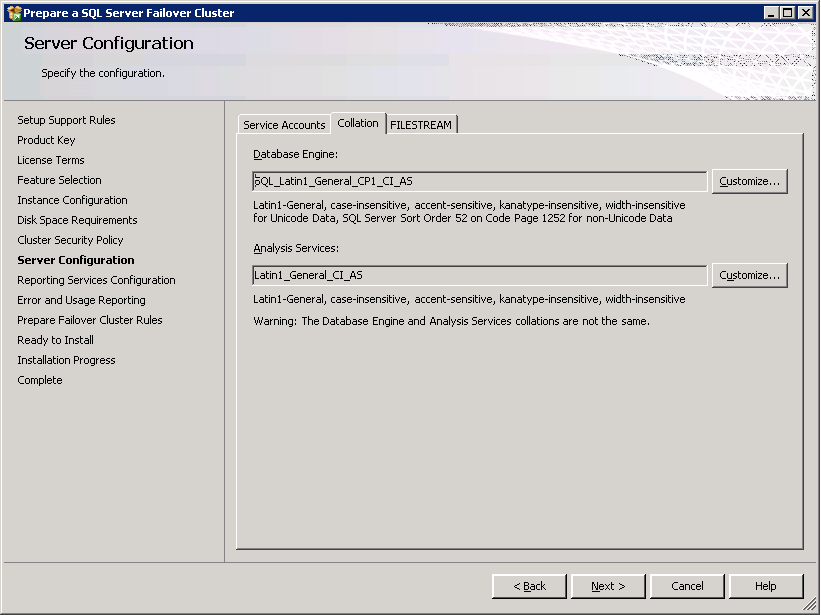
You can assign the same login account to all SQL Server services, or you can configure each service account individually. The startup type is set to manual for all cluster-aware services, including full-text search and SQL Server Agent, and cannot be changed during installation. Microsoft recommends that you configure service accounts individually to provide least privileges for each service, where SQL Server services are granted the minimum permissions they have to have complete their tasks. For more information, see [SQL Server Configuration - Service Accounts](http://technet.microsoft.com/en-us/library/cc281953.aspx) (http://technet.microsoft.com/en-us/library/cc281723.aspx) and [Setting Up Windows Service Accounts](http://technet.microsoft.com/en-us/library/ms143504.aspx) (http://technet.microsoft.com/en-us/library/ms143504.aspx).

To specify the same login account for all service accounts in this instance of SQL Server, provide credentials in the fields at the bottom of the page.

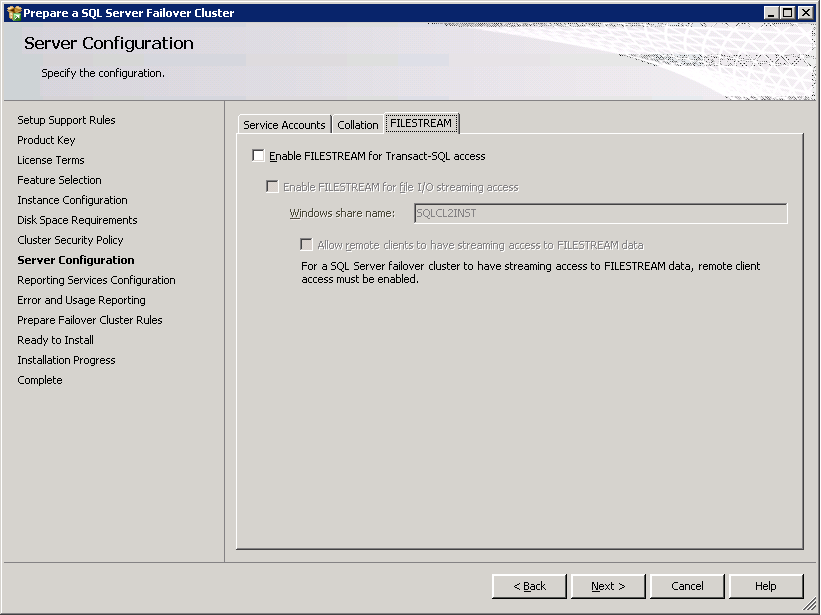
**Security note**: Do not use a blank password. Use a strong password.

When you are finished specifying login information for SQL Server services, click **Next**.

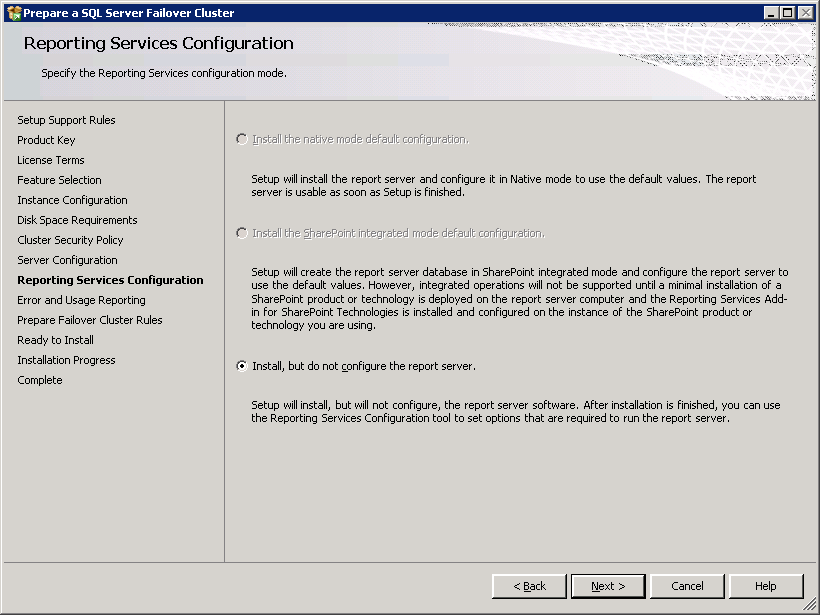
1. Use the **Collation** tab to specify nondefault collations for the Database Engine and Analysis Services.



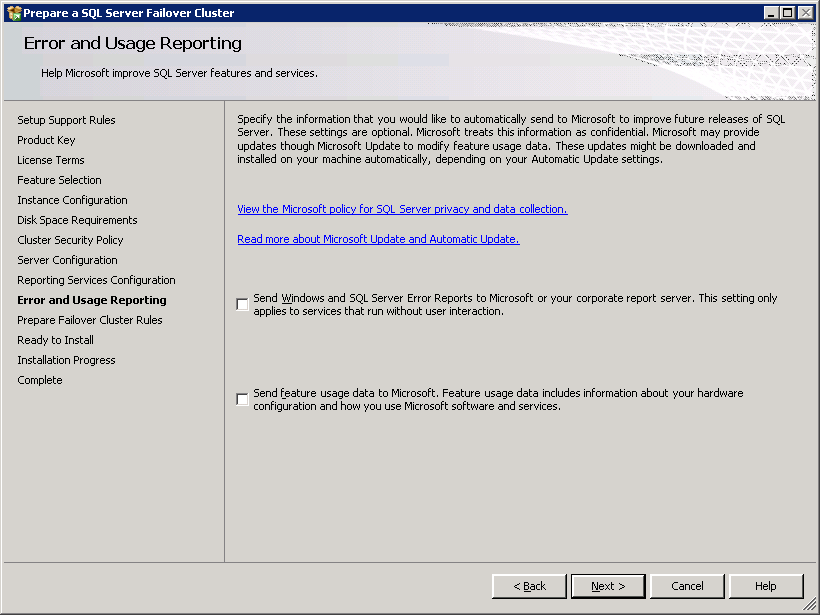
1. Use the **FILESTREAM** tab to enable FILESTREAM for your instance of SQL Server. Click **Next** to continue.



1. Use the Reporting Services Configuration page to specify the kind of Reporting Services installation to create. For failover cluster installation, the option is set to “Install, but do not configure the report server.” You must configure Reporting Services after you complete the installation.



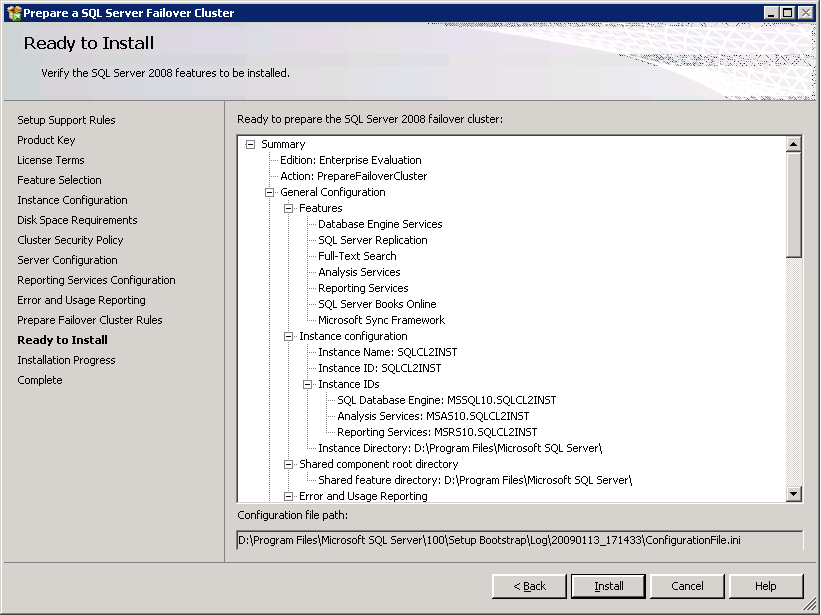
1. On the Error and Usage Reporting page, specify the information that you want to send to Microsoft that will help improve SQL Server. By default, options for error reporting and feature usage are enabled.



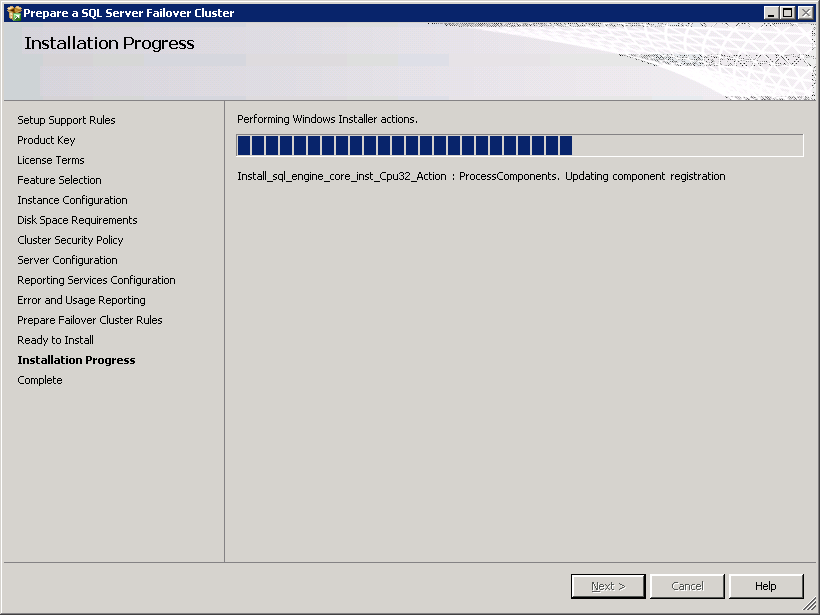
1. The System Configuration Checker runs one more set of rules to validate your configuration with the SQL Server features that you have specified.



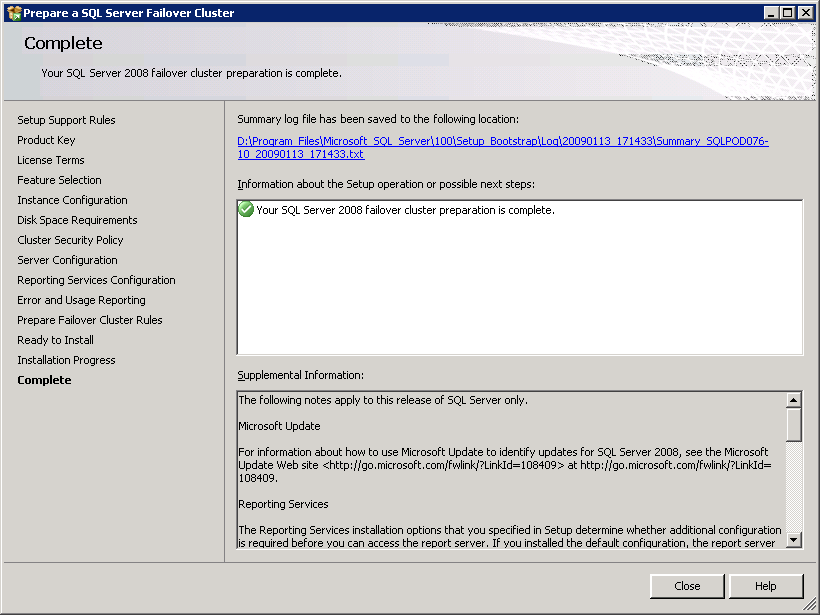
1. The Ready to Install page displays a tree view of installation options that you specified during setup. To continue, click **Install**.



1. During installation, the Installation Progress page provides status so that you can monitor installation progress as Setup continues.



1. After installation, the Complete page provides a link to the summary log file for the installation and other important notes.



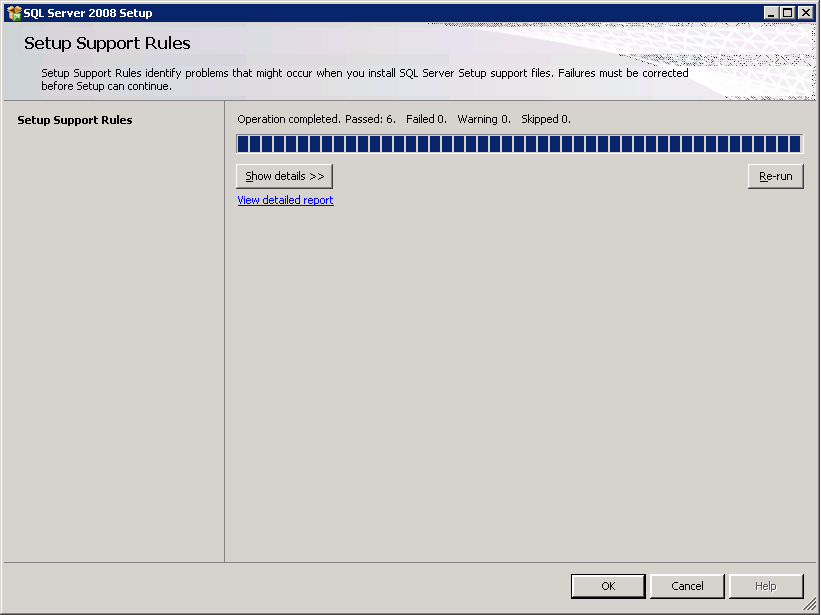
1. To complete the SQL Server installation process, click **Close**.
2. If you are instructed to restart the computer, do so now. It is important to read the message from the Installation Wizard when you have finished with setup. For information about setup log files, see [How To: View SQL Server Setup Log Files](http://msdn.microsoft.com/en-us/library/ms143702.aspx) (http://msdn.microsoft.com/en-us/library/ms143702.aspx).
3. Repeat the previous steps to prepare the other nodes for the failover cluster. You can also use the autogenerated configuration file to run prepare on the other nodes. For more information, see [How to: Install SQL Server 2008 Using a Configuration File](http://msdn.microsoft.com/en-us/library/dd239405.aspx) (http://msdn.microsoft.com/en-us/library/dd239405.aspx).

## Advanced/Enterprise Failover Cluster Installation Step 2: Complete

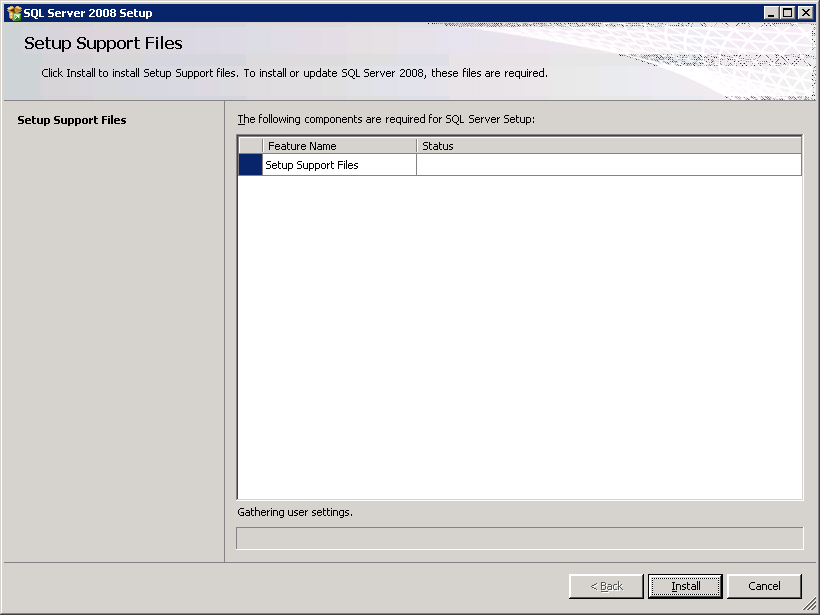
1. After preparing all the nodes as described in step 1, run Setup on one of the prepared nodes, preferably the one that owns the shared disk. On the Advanced page of the SQL Server Installation Center, click **Advanced cluster completion**.



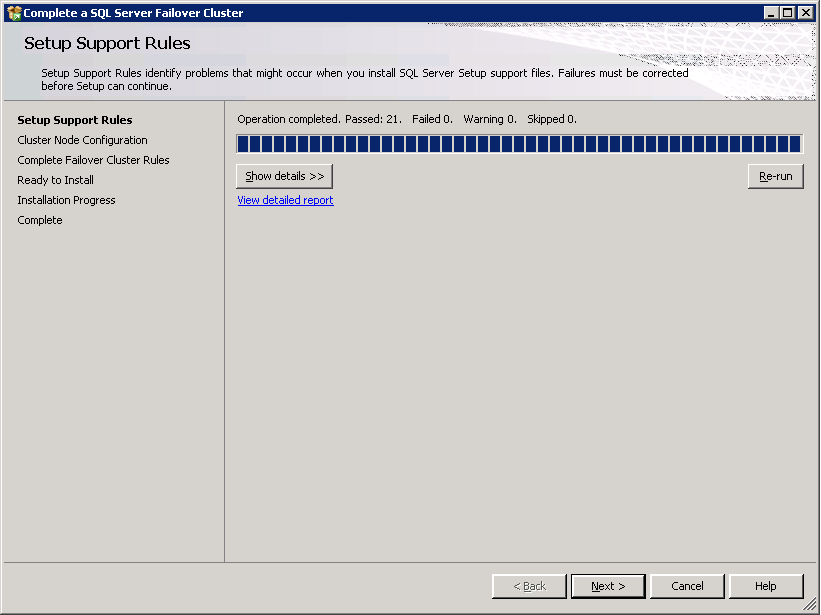
1. The System Configuration Checker runs a discovery operation on your computer. To continue, click **OK**. You can view the details on the screen by clicking **Show Details**, or as an HTML report by clicking **View detailed report**.



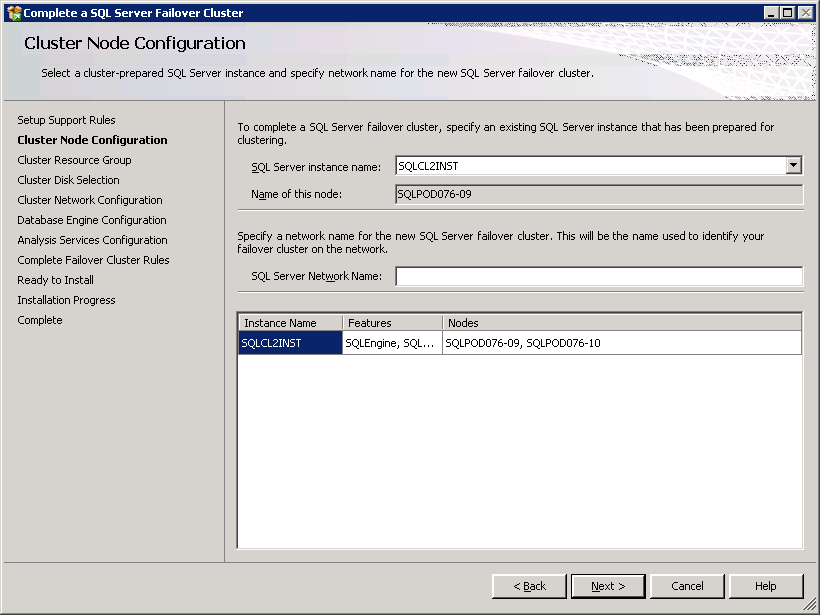
1. On the Setup Support Files page, click **Install** to install the setup support files.



1. The System Configuration Checker verifies the system state of your computer before setup continues. After the check is complete, click **Next** to continue. You can view the details on the screen by clicking **Show Details**, or as an HTML report by clicking **View detailed report**.



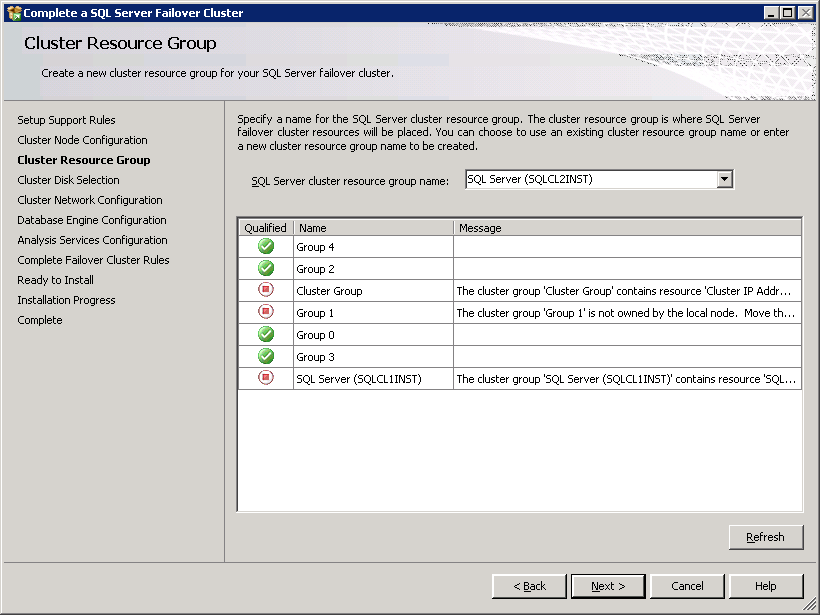
1. Use the Cluster Node Configuration page to select the instance name prepared for clustering, and specify the network name for the new SQL Server failover cluster. This is the name that is used to identify your failover cluster on the network.



**Note:** This is known as the virtual SQL Server name in earlier versions of SQL Server failover clusters.

1. Use the Cluster Resource Group page to specify the cluster resource group name where SQL Server virtual server resources will be located. To specify the SQL Server cluster resource group name. You have two options:

* Use the list to specify an existing group to use.
* Type the name of a new group to create.



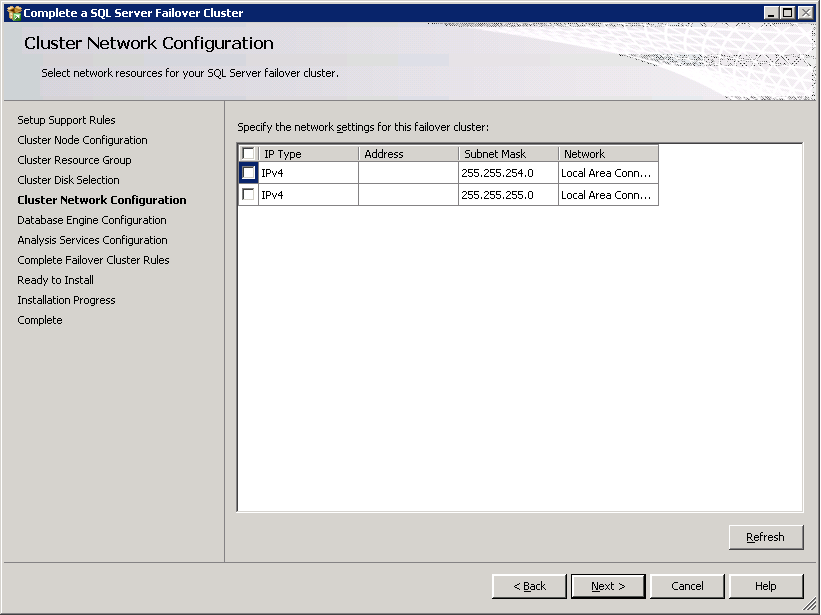
1. On the Cluster Disk Selection page, select the shared cluster disk resource for your SQL Server failover cluster. The cluster disk is where the SQL Server data will be put. More than one disk can be specified. **Available shared disks** displays a list of available disks, whether each is qualified as a shared disk, and a description of each disk resource. Click **Next** to continue.



**Note:** The first drive is used as the default drive for all databases, but can be changed on the Database Engine or Analysis Services configuration pages.

1. On the Cluster Network Selection page, specify the network resources for your failover cluster instance:

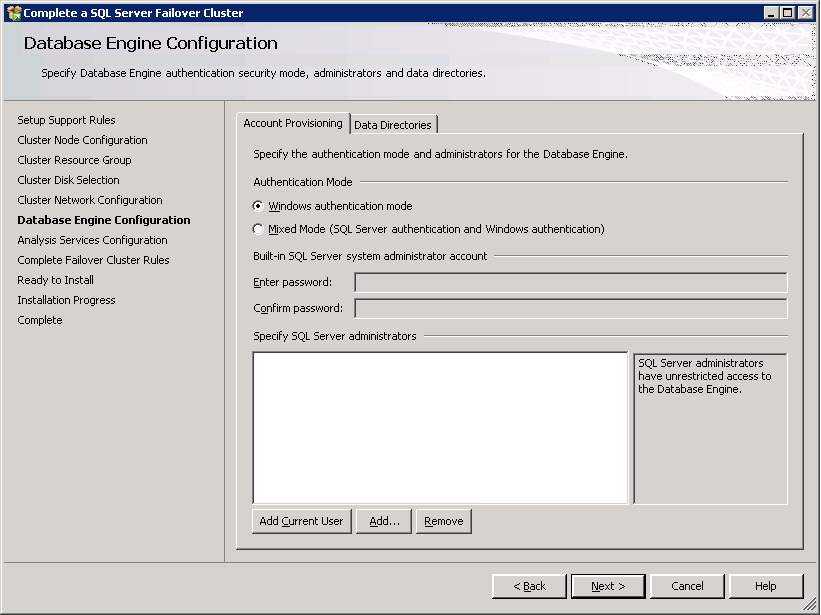
* Network settings — Specify the IP type and IP address for your failover cluster instance. On Windows Server 2008 failover clusters, SQL Server supports the use of DHCP addresses. Before choosing this option, make sure that any network security such as firewalls and IPsec in use between this SQL Server and its clients can accommodate server-side DHCP. For example, some firewalls will require a static port for proper configuration of the firewall.



Click **Next** to continue.

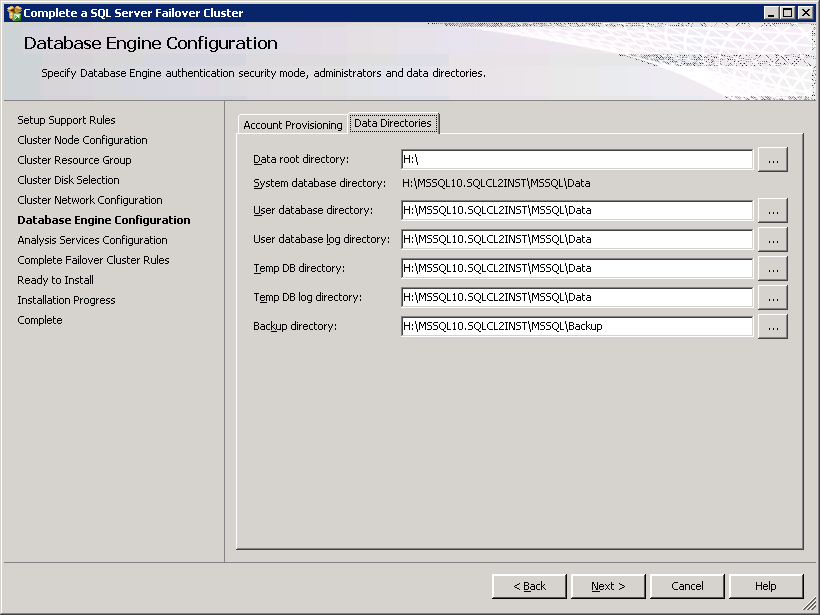
Work flow for the rest of this procedure depends on the features that you have specified for your installation. You might not see all the pages, depending on your selections.

1. On the Database Engine Configuration page, use the **Account Provisioning** tab to specify the following:
   * Authentication mode - select Windows authentication or Mixed Mode authentication for your instance of SQL Server. If you select Mixed Mode authentication, you must provide a strong password for the built-in SQL Server system administrator account.  
       
     After a device establishes a successful connection to SQL Server, the security mechanism is the same for both Windows authentication and Mixed Mode. For more information, see [Database Engine Configuration - Account Provisioning](http://technet.microsoft.com/en-us/library/cc281849.aspx) (http://technet.microsoft.com/en-us/library/cc281849.aspx).
   * SQL Server administrators - you must specify at least one system administrator for the instance of SQL Server. To add the account under which SQL Server Setup is running, click **Add Current User**. To add or remove accounts from the list of system administrators, click **Add** or **Remove**, and then edit the list of users, groups, or computers that will have administrator privileges for the instance of SQL Server. For more information, see [Database Engine Configuration - Account Provisioning](http://technet.microsoft.com/en-us/library/cc281849.aspx) (http://technet.microsoft.com/en-us/library/cc281849.aspx).



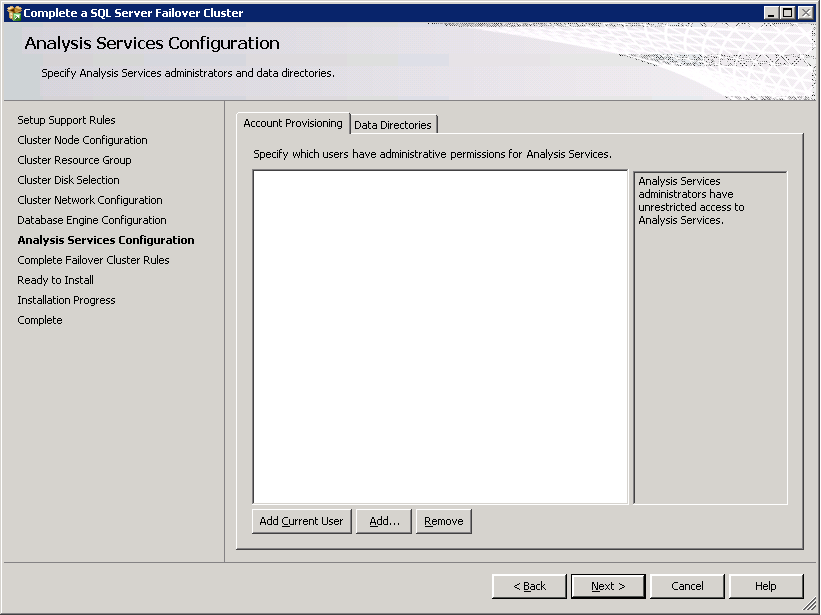
When you are finished editing the list, click **OK**. Verify the list of administrators in the configuration dialog box. When the list is complete, click **Next**.

1. Use the **Data** **Directories** tab to specify nondefault installation directories. To install to default directories, click **Next**.



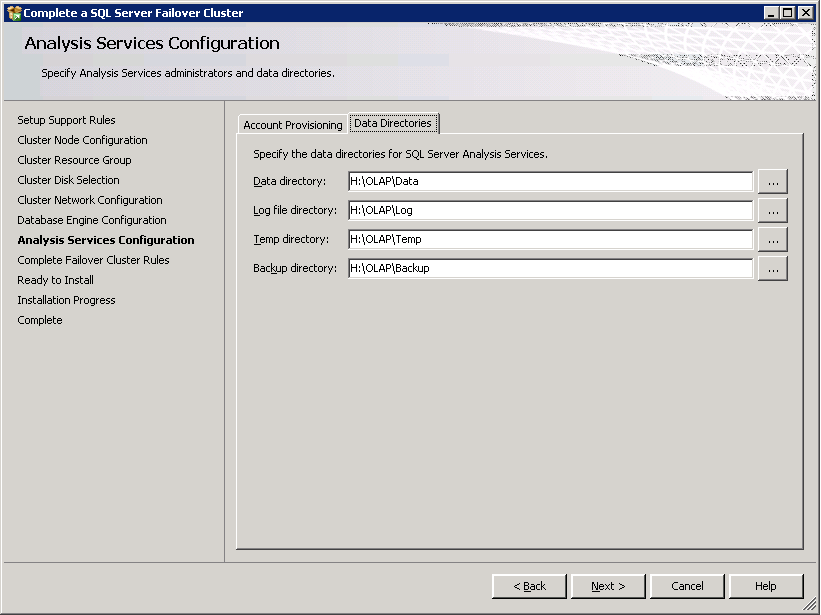
**Important:** If you specify nondefault installation directories, make sure that the installation folders are unique to this instance of SQL Server. None of the directories in this tab should be shared with directories from other instances of SQL Server. The data directories should be located on the shared cluster disk for the failover cluster.

1. Use the **Account Provisioning** tab to specify users or accounts that will have administrator permissions for Analysis Services. You must specify at least one system administrator for Analysis Services. To add the account under which SQL Server Setup is running, click **Add Current User**. To add or remove accounts from the list of system administrators, click **Add** or **Remove**, and then edit the list of users, groups, or computers that will have administrator privileges for Analysis Services. For more information, see [Analysis Services Configuration - Account Provisioning](http://technet.microsoft.com/en-us/library/cc281723.aspx) (http://technet.microsoft.com/en-us/library/cc281723.aspx).



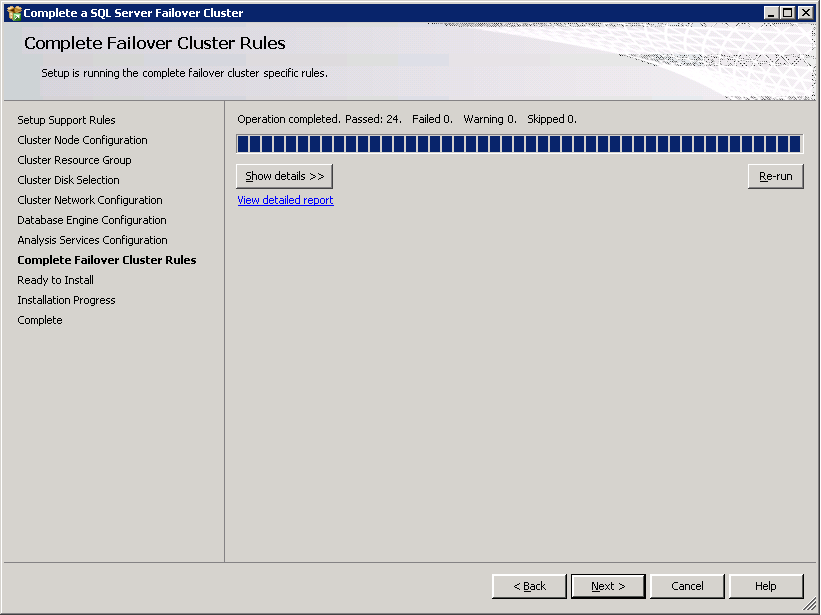
When you are finished editing the list, click **OK**. Verify the list of administrators in the configuration dialog box. When the list is complete, click **Next**.

1. Use the **Data Directories** tab to specify nondefault installation directories. To install to default directories, click **Next**.

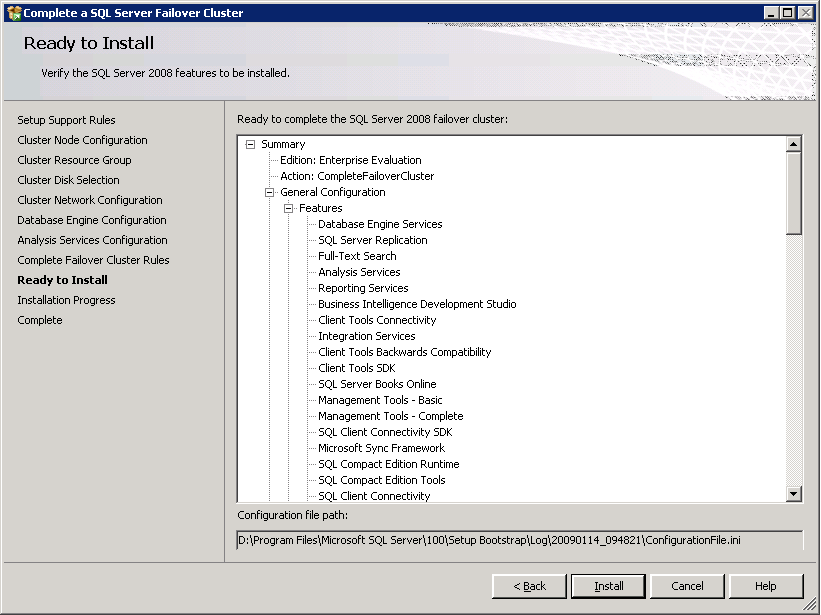


**Important:** If you specify nondefault installation directories, make sure that the installation folders are unique to this instance of SQL Server. None of the directories in this tab should be shared with directories from other instances of SQL Server. The data directories should be located on the shared cluster disk for the failover cluster.

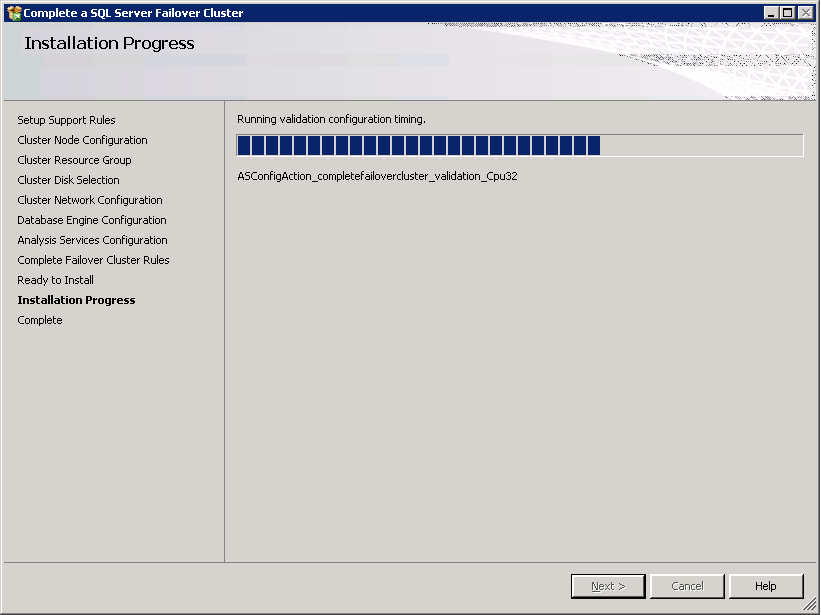
1. The System Configuration Checker runs one more set of rules to validate your configuration with the SQL Server features that you have specified.



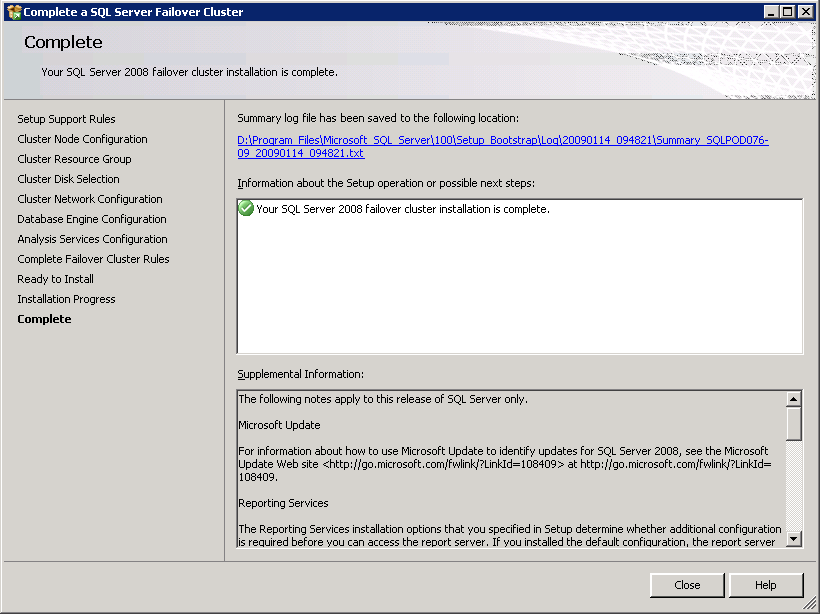
1. The Ready to Install page displays a tree view of installation options that you specified during setup. To continue, click **Install**.



1. During installation, the Installation Progress page provides status so that you can monitor installation progress as Setup continues.



1. After installation, the Complete page provides a link to the summary log file for the installation and other important notes. To complete the SQL Server installation process, click **Close**.



With this step, all the prepared nodes for the same failover cluster are now part of the completed SQL Server failover cluster.

# Appendix C: Add Node

Use this procedure to manage nodes for an existing SQL Server failover cluster instance.

**Important:** To update or remove a SQL Server failover cluster, you must be a local administrator with permission to log in as a service on all nodes of the failover cluster. For local installations, you must run Setup as an administrator. If you install SQL Server from a remote share, you must use a domain account that has read and execute permissions on the remote share.

Setup does not install .NET Framework 3.5 SP1 on a clustered operating system. You must install .NET Framework 3.5 SP1 before you run Setup.

You may need to apply cumulative updates to the original media before you install SQL Server 2008, if you are affected by a known issue in the Setup program. For more information about known issues and detailed instructions, see [How to update or slipstream an installation SQL Server 2008](http://support.microsoft.com/kb/955392) (http://support.microsoft.com/kb/955392).

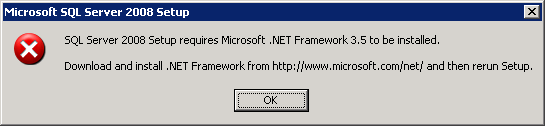
Note that setup operations for SQL Server failover clustering have changed in this release. To install or upgrade a SQL Server failover cluster, you must run the Setup program on each node of the failover cluster.

To add a node to an existing SQL Server failover cluster, you must run SQL Server Setup on the node that is to be added to the SQL Server failover cluster instance. Do not run Setup on the active node.

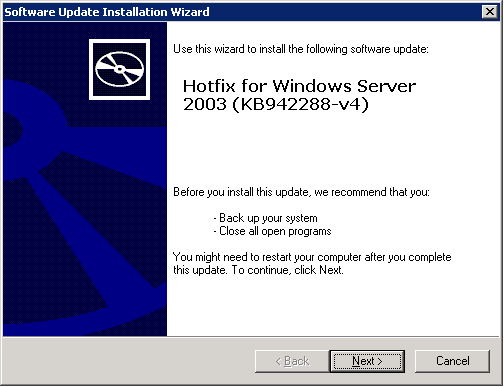
To remove a node from an existing SQL Server failover cluster, you must run SQL Server Setup on the node that is to be removed from the SQL Server failover cluster instance.

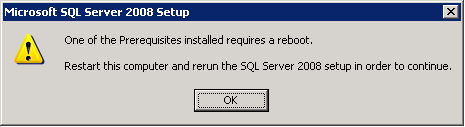
## To Add a Node to an Existing SQL Server 2008 Failover Cluster

1. Insert the SQL Server installation media, and from the root folder, double-click Setup.exe. To install from a network share, navigate to the root folder on the share, and then double-click Setup.exe. You may be asked to install the prerequisites if they are not previously installed.

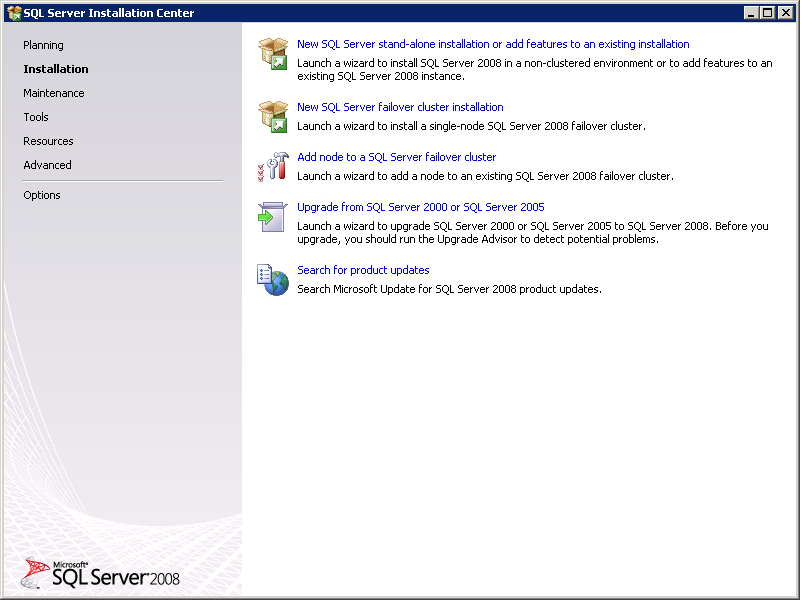


Install Windows Installer 4.5. Windows Installer 4.5 is required, and it can be installed by the Installation Wizard. If you are prompted to restart your computer, restart, and then start SQL Server 2008 Setup.exe again.

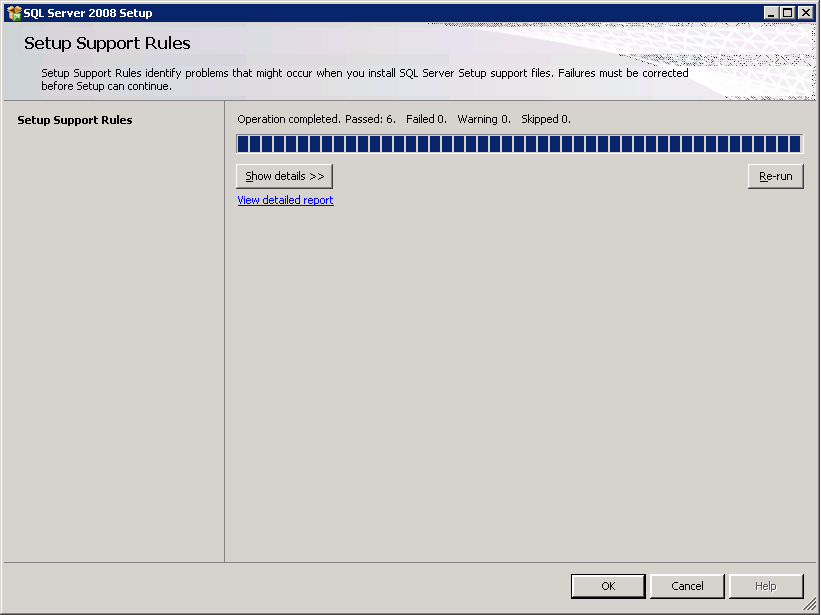




1. When prerequisites are installed, the Installation Wizard will launch the SQL Server Installation Center. To add a node to an existing failover cluster instance, click **Installation** in the left-hand pane. Then, click **Add node to a SQL Server failover cluster**.

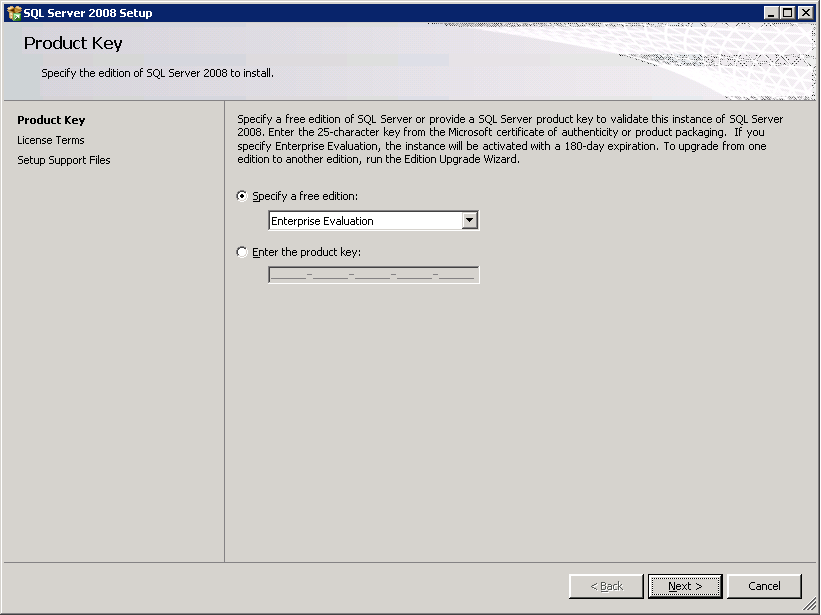


The System Configuration Checker will run a discovery operation on your computer. To continue, click **OK**. Setup log files have been created for your installation. For more information about log files, see [How To: View SQL Server Setup Log Files](http://msdn.microsoft.com/en-us/library/ms143702.aspx) (http://msdn.microsoft.com/en-us/library/ms143702.aspx).

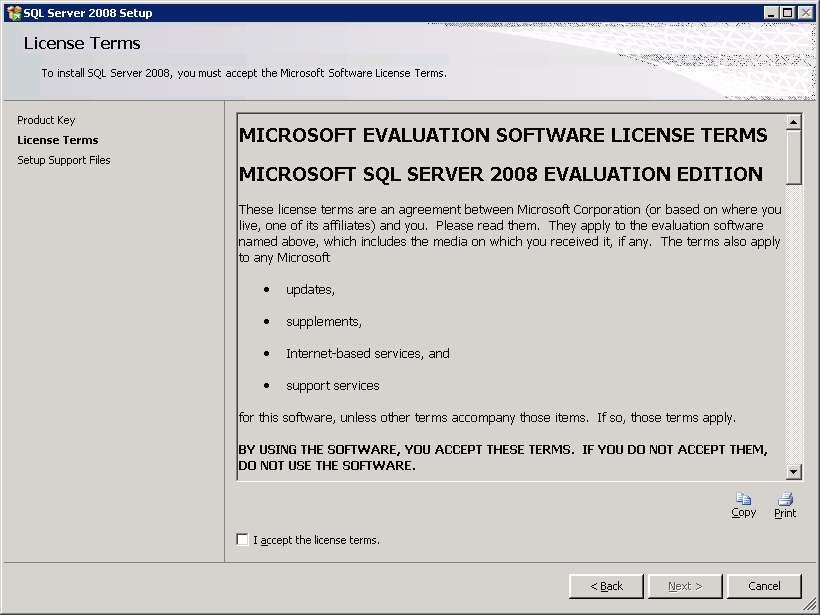


1. On the Product Key page, specify the PID key for a production version of the product. Note that the product key you enter for this installation must be for the same SQL Server 2008 edition as that which is installed on the active node.

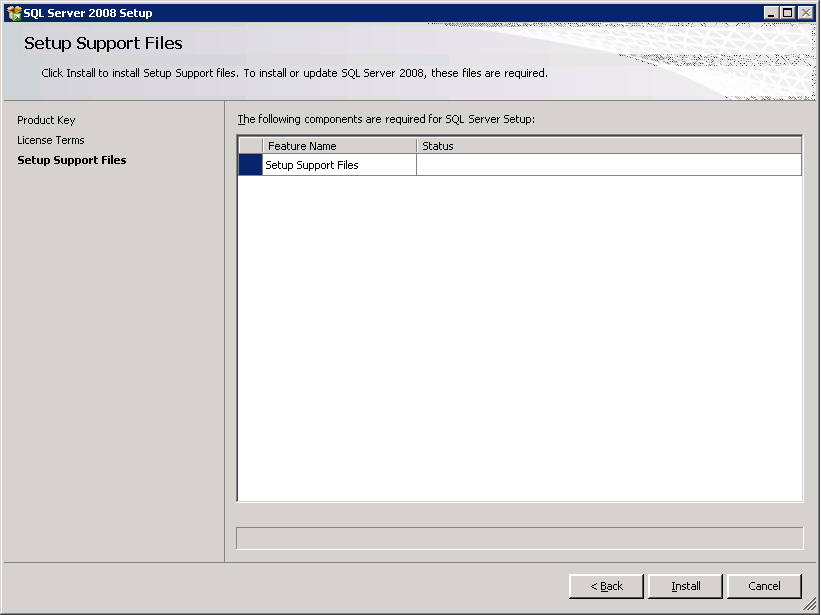
**Note**: The Product Key and License Terms pages show up after the setup support files page if you already installed the setup support files during a previous installation.



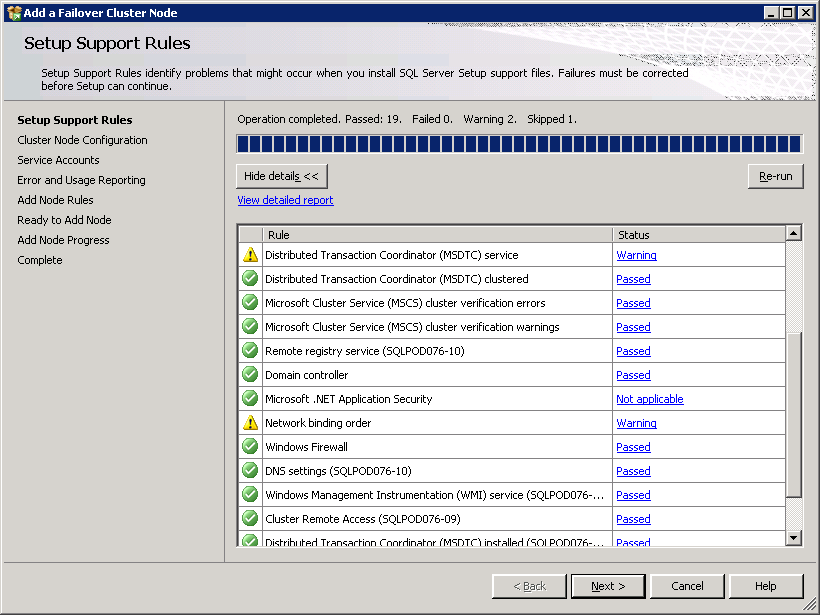
1. On the License Terms page, read the license agreement, and then select the check box to accept the licensing terms and conditions. To continue, click **Next**. To end Setup, click **Cancel**.



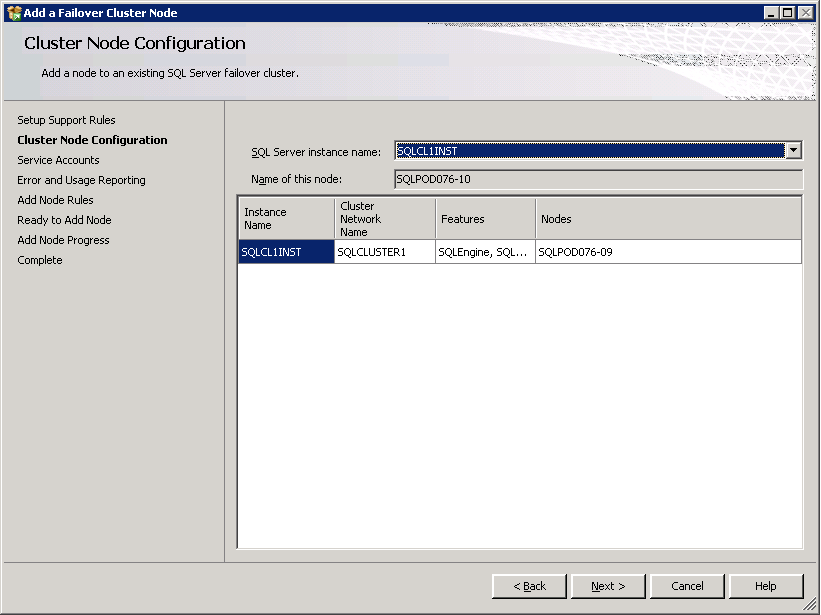
1. On the Setup Support Files page, click **Install** to install the setup support files. To install prerequisites, click **Install**.



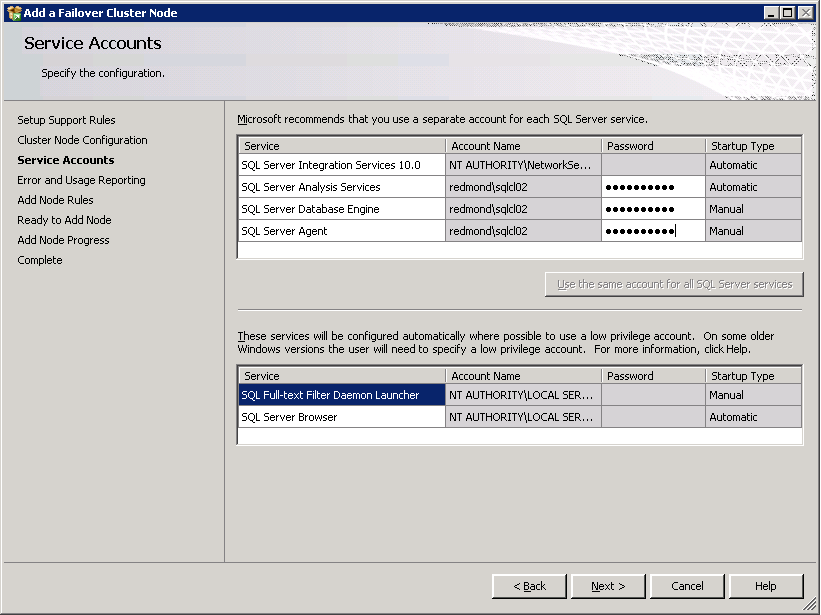
1. The System Configuration Checker will verify the system state of your computer before Setup continues. After the check is complete, click **Next** to continue.



1. On the Cluster Node Configuration page, use the **SQL Server instance name** box to specify the name of the SQL Server 2008 failover cluster instance that will be modified during this setup operation.



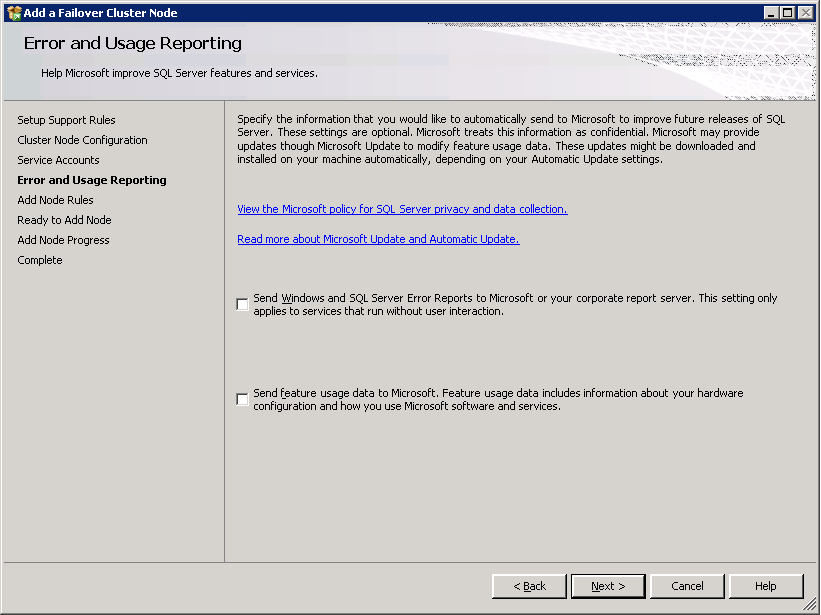
1. On the Service Accounts page, specify login accounts for SQL Server services. The actual services that are configured on this page depend on the features you selected to install. For failover cluster installations, account name and startup type information will be prepopulated on this page based on settings provided for the active node. You must provide passwords for each account.



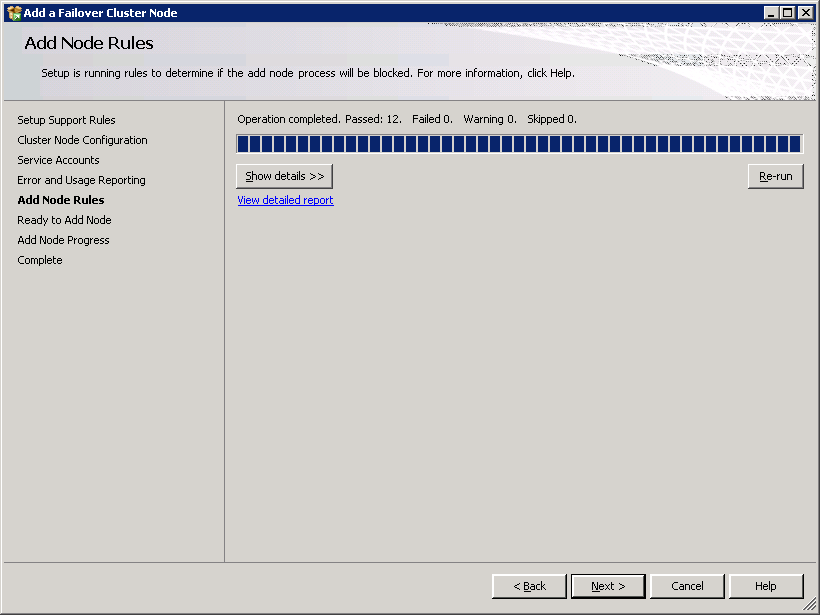
**Security note**: Do not use a blank password. Use a strong password.

When you are finished specifying login information for SQL Server services, click **Next**.

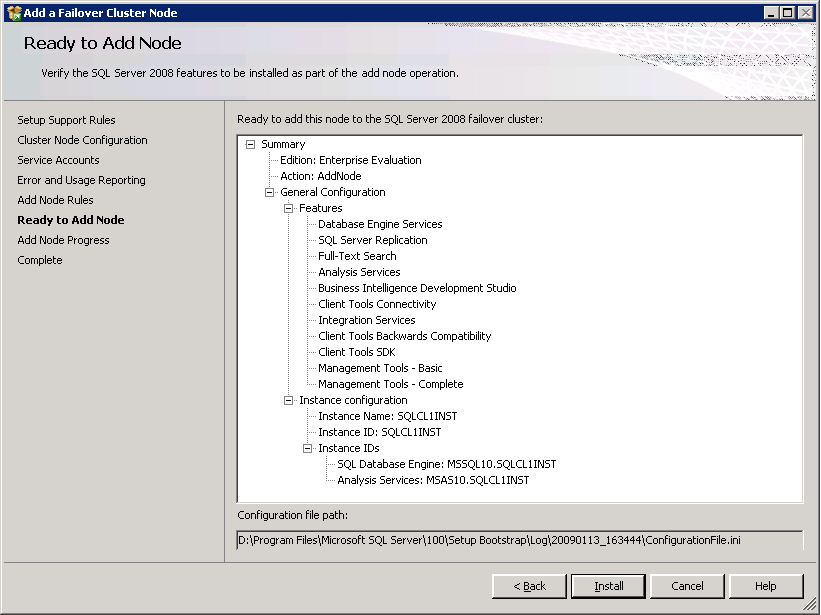
1. On the Error and Usage Reporting page, specify the information to send to Microsoft that will help to improve SQL Server. By default, options for error reporting and feature usage are enabled.



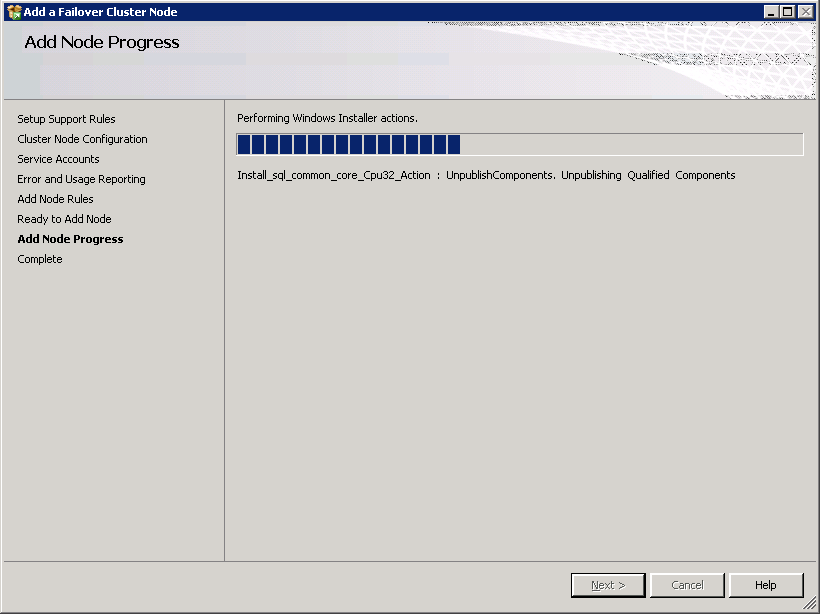
1. The System Configuration Checker will run one more set of rules to validate your computer configuration with the SQL Server features you have specified.



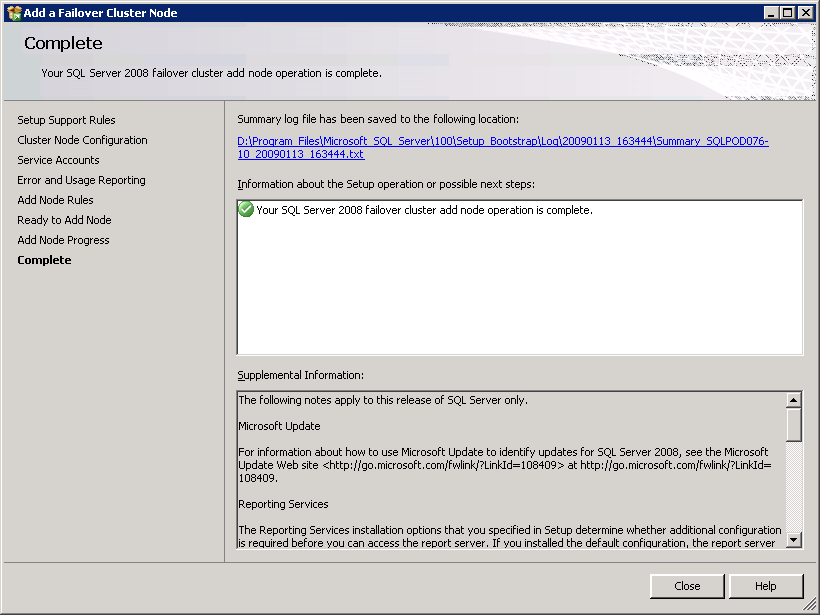
1. The Ready to Add Node page displays a tree view of installation options that you specified during setup.



1. The Add Node Progress page provides status so you can monitor add node progress as Setup proceeds.



1. After installation, the Complete page provides a link to the summary log file for the installation and other important notes. To complete the SQL Server installation process, click **Close**.



1. If you are instructed to restart the computer, do so now. It is important to read the message from the Installation Wizard when you are done with setup. For more information about setup log files, see [How To: View SQL Server Setup Log Files](http://msdn.microsoft.com/en-us/library/ms143702.aspx) (http://msdn.microsoft.com/en-us/library/ms143702.aspx).

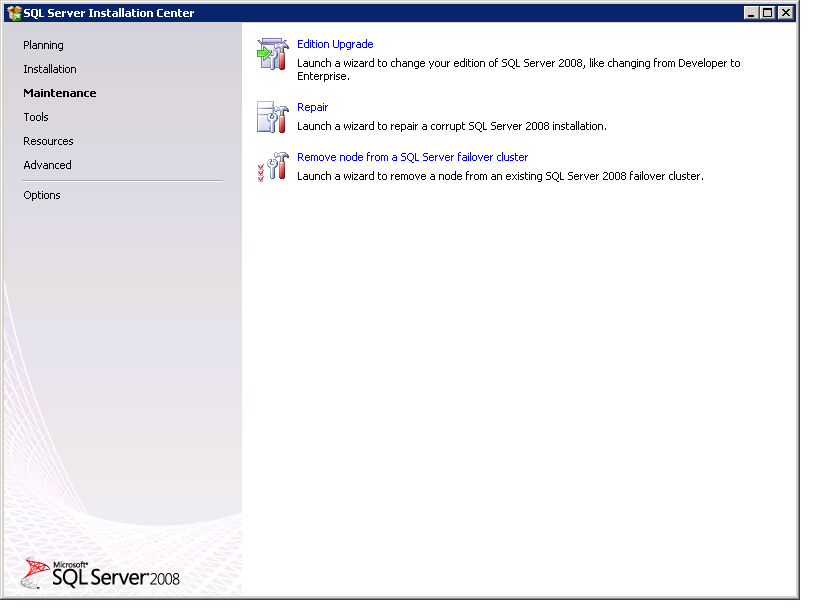
# Appendix D: Removing a Node

Removing a node makes that node unavailable to the specific instance of SQL Server the **RemoveNode** action was run against. If you must remove a node that is currently hosting your instance of SQL Server, note that your SQL instance will fail over to another node of the cluster. As a best practice, you should remove nodes that are not running the active SQL Server instance, to avoid this failover scenario. If the node to be removed is currently running the SQL Server failover cluster instance that **RemoveNode** is executed on, consider moving the group to a node that will not be removed from the cluster during a maintenance window. This can help maintain the maximum uptime and avoid unplanned failovers.

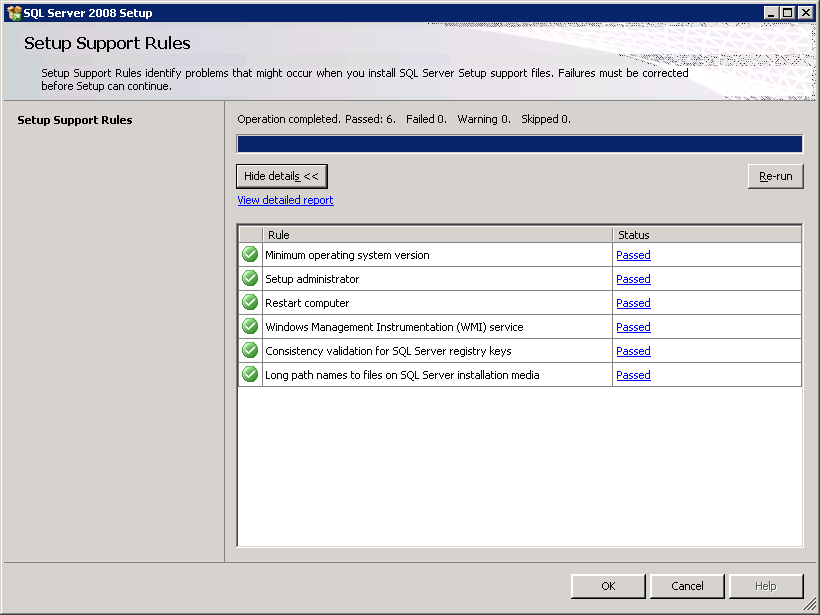
**Note:** Instances that did not have the **RemoveNode** part of setup run against them will not be affected by the operation.

## To Remove a Node from an Existing SQL Server 2008 Failover Cluster

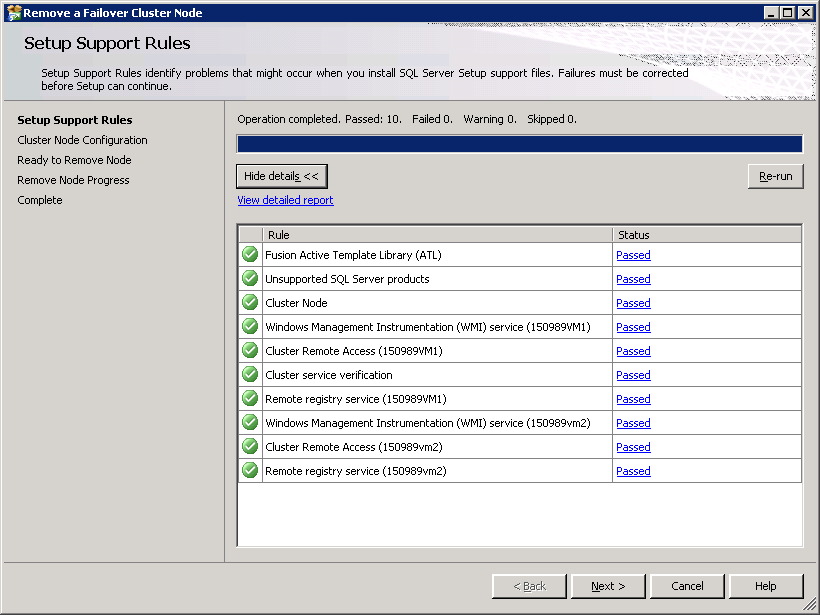
1. Insert the SQL Server installation media. From the root folder, double-click Setup.exe. To install from a network share, navigate to the root folder on the share, and then double-click Setup.exe. The Installation Wizard will launch the SQL Server Installation Center. To remove a node from an existing failover cluster instance, click **Maintenance** in the left-hand pane, and then click **Remove node from a SQL Server failover cluster**.



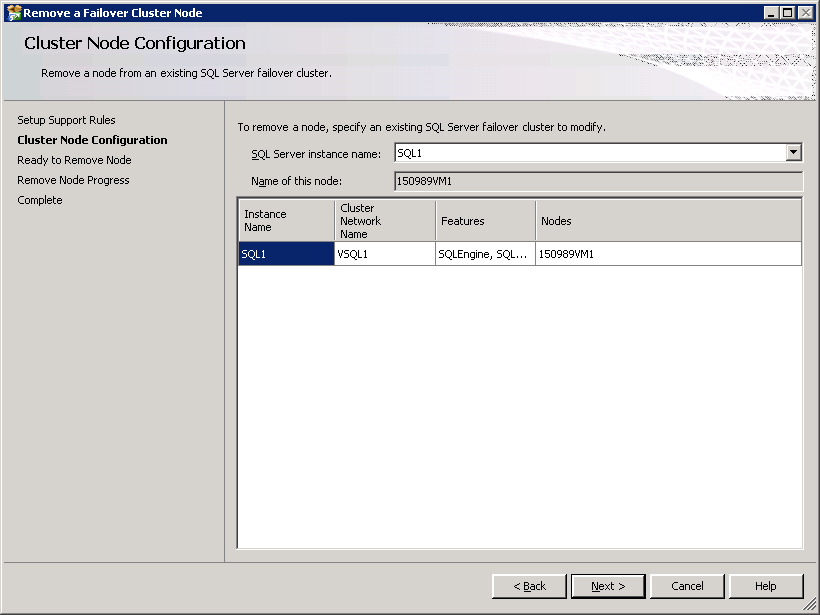
1. The System Configuration Checker will run a discovery operation on your computer. To continue, click **OK**. Setup log files have been created for your installation. For more information about log files, see [How To: View SQL Server Setup Log Files](http://msdn.microsoft.com/en-us/library/ms143702.aspx) (http://msdn.microsoft.com/en-us/library/ms143702.aspx).



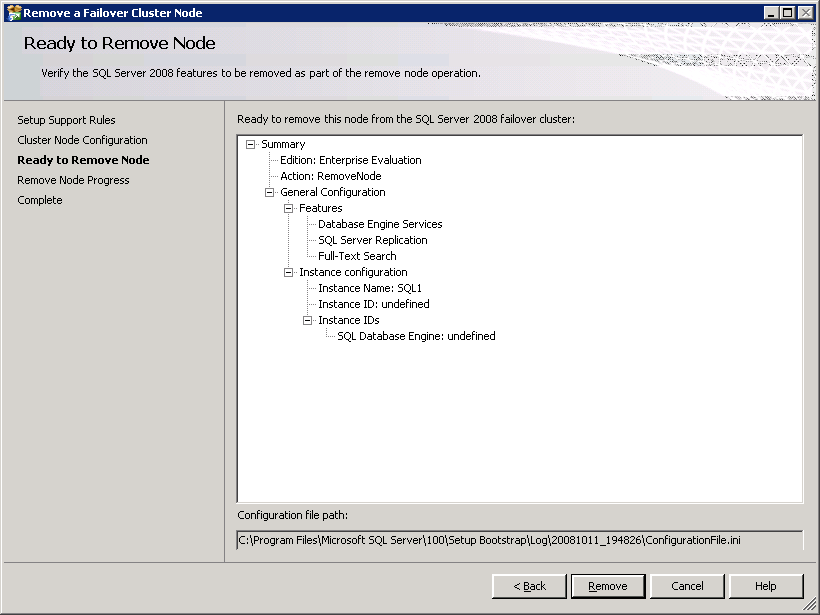
1. The System Configuration Checker will verify the system state of your computer before setup continues. After the check is complete, click **Next** to continue.



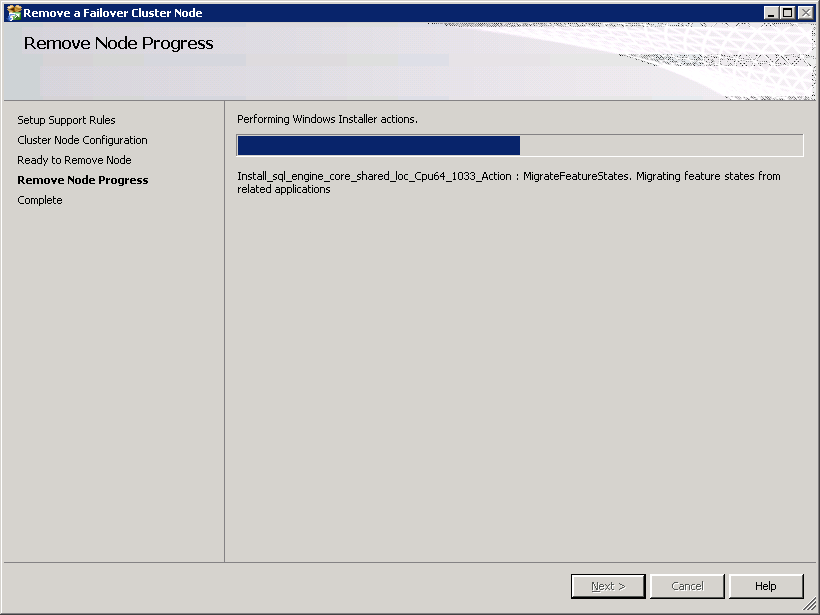
1. On the Cluster Node Configuration page, use the drop-down box to specify the name of the SQL Server 2008 failover cluster instance that will be modified during this setup operation. The node that will be removed will be listed in **Name of this node**.



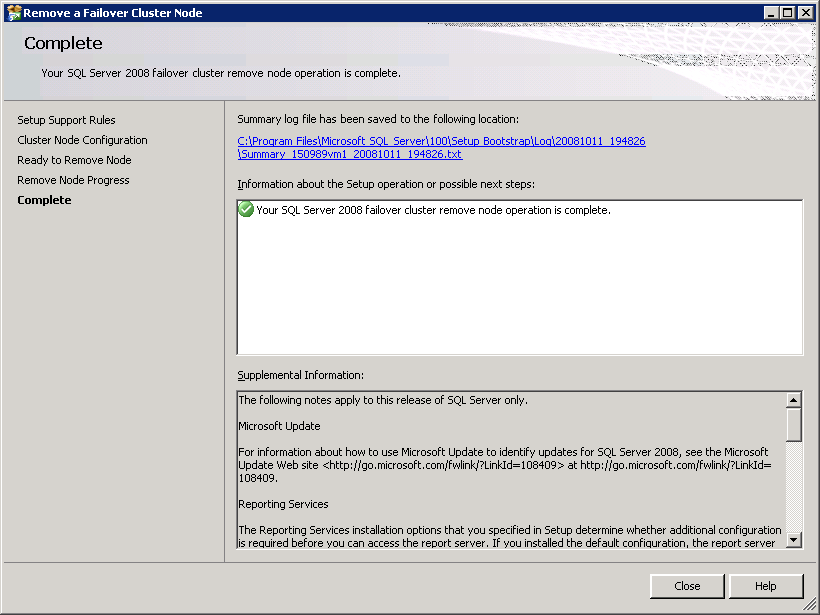
1. The Ready to Remove page displays a tree view of options that will be removed during setup. To continue, click **Remove**.



1. During the remove operation, the Remove Node Progress page provides status.



1. The Complete page provides a link to the summary log file for remove node and other important notes. To complete the SQL Server remove node, click **Close**. For more information about setup log files, see [How To: View SQL Server Setup Log Files](http://msdn.microsoft.com/en-us/library/ms143702.aspx) (http://msdn.microsoft.com/en-us/library/ms143702.aspx).



# Appendix E: SQL Server 2008 Failover Cluster Setup - Troubleshooting Error Conditions

The first step in troubleshooting a setup error is to locate the setup action where it occurred. The following is a quick summary of how to locate the action that had an error or errors which caused Setup to fail.

**Note:** Setup logs are located at on each cluster node under the following folder: %programfiles%\Microsoft SQL Server\100\Setup Bootstrap\Log

1. First check the Summary file to locate the exit code of Setup, and to see if any features failed. Here is an example.

Overall summary:

Final result: Failed: see details below

Exit code (Decimal): -2068643839

Exit facility code: 1203

Exit error code: 1

Exit message: Failed: see details below

Start time: 2008-05-24 04:39:44

End time: 2008-05-24 04:47:34

Requested action: InstallFailoverCluster

Log with failure: ...

Exception help link: ...

...

Detailed results:

Feature: Database Engine Services

Status: Passed

MSI status: Passed

Configuration status: Passed

...

Feature: Client Tools Connectivity

Status: Failed: see logs for details

MSI status: Failed: see details below

MSI error code: -2147483648

MSI log file location: ...

MSI error description: A later version of Microsoft Visual Studio Tools for Applications 2.0 - ENU is already installed.

Configuration status: Passed

1. If there are failed features, you can locate the actions that failed by searching for the following in the \*\_Detail log file.

*failed during execution*

1. If no features are listed as being failed, it probably means there was an exception that caused Setup to exit.
2. Locate the failed action by searching for the following in the \*\_Detail log file.

*threw an exception during execution*

## Error Messages and Troubleshooting Guidance

In the case of validation errors, the Summary log file will contain the error message specific to the problematic input setting.

In the following table, the scenarios represent the choice made on the Setup landing page. In the case of command line install, the scenario represents the value passed into the /ACTION parameter.

| Error message | Scenarios | Troubleshooting tips |
| --- | --- | --- |
| The specified disk resource '%1' cannot be used as a failover cluster disk. Reason: %2 | INSTALLFAILOVERCLUSTER  COMPLETEFAILOVERCLUSTER | * Applies to the /FAILOVERCLUSTERDISKS setting * The disk qualification code failed to qualify the disk * Make sure the type of the disk is not “Local Quorum” * Make sure the disk has no dependencies * Make sure nothing depends on the disk * Make sure the disk is online * Make sure the local node owns the drive * If other drives are mounted onto the disk, make sure each drive that is mounted is also qualified for the local node |
| The specified disk resource '%1' is not a valid cluster physical disk resource name. To continue, specify a valid disk resource and retry. | INSTALLFAILOVERCLUSTER  COMPLETEFAILOVERCLUSTER | * Applies to the /FAILOVERCLUSTERDISKS setting * Fails when Setup cannot find the shared disk specified * If you are manually editing an .ini file and multiple disks are to be specified, the disks referenced must be contained within “” and separated with a space as the delimiter |
| There were no cluster disks specified. To continue, specify at least one disk to create a SQL Server failover cluster instance. | INSTALLFAILOVERCLUSTER  COMPLETEFAILOVERCLUSTER | * Applies to the /FAILOVERCLUSTERDISKS setting * Occurs when the user doesn’t specify the setting and Setup couldn’t find an available shared disk |
| No shared disk cluster resources could be found for the SQL Server failover cluster instance name '%1'. | UPGRADE  REPAIR  REBUILDDATABASE | * Applies to the /FAILOVERCLUSTERDISKS setting * Occurs when Setup cannot find any shared disk resources in the existing failover cluster group |
| The cluster group name cannot be null or empty. To continue, specify a valid group name and retry, or remove the command line parameter so Setup can provide a default value. | INSTALLFAILOVERCLUSTER  COMPLETEFAILOVERCLUSTER | * Applies to the /FAILOVERCLUSTERGROUP setting * Happens only if the setup configuration was unable to calculate a group name. Refer to the detail log for any errors that may have occurred during setup |
| The cluster group cannot be determined for the instance name '%1'. This indicates there is a problem with the product registry setting for ClusterName, with product discovery, or the cluster resources. | ADDNODE  UPGRADE  REPAIR  REBUILDDATABASE | * Applies to the /FAILOVERCLUSTERGROUP setting * Happens if Setup could not locate the group based on the instance name being operated on |
| The cluster group '%1' could not be found. | ADDNODE  UPGRADE  REPAIR  REBUILDDATABASE | * Applies to the /FAILOVERCLUSTERGROUP setting * If specified by the user, the group named cannot be found in the cluster * If the setting was not specified, it means the calculated name of the group does not map to an existing group in the cluster, so the user must specify the group name |
| The Cluster Group %1 is not owned by the node where SQL Server Setup is running. To continue, run Setup again on the node which owns the group. | REBUILDDATABASE | * Applies to the /FAILOVERCLUSTERGROUP setting * For a clustered instance, the rebuild database scenario must be run on the owning node of the cluster group |
| The cluster group '%1' is reserved by Windows Failover Clustering and cannot be used as a SQL group. | INSTALLFAILOVERCLUSTER  COMPLETEFAILOVERCLUSTER | * Applies to the /FAILOVERCLUSTERGROUP setting * Check the name of the group that is attempted to be used * Reserved groups on Windows Server 2008 like Available Storage cannot be used for a SQL Server failover instance |
| The cluster group '%1' is not owned by the local node. Move the group to the local node to continue. | INSTALLFAILOVERCLUSTER  COMPLETEFAILOVERCLUSTER | * Applies to the /FAILOVERCLUSTERGROUP setting * An existing group that is not owned by the local machine was attempted to be used for the failover cluster |
| The cluster group '%1' is not online. Bring the group online to continue. | INSTALLFAILOVERCLUSTER  COMPLETEFAILOVERCLUSTER | * Applies to the /FAILOVERCLUSTERGROUP setting * An existing group that is offline was attempted to be used for the failover cluster * Bring the group online and run Setup again |
| The cluster group '%1' contains resource '%2' of type '%3' that are not permitted in a SQL cluster group. Ensure the cluster group does not contain SQL resources from another instance or Generic Service types. | INSTALLFAILOVERCLUSTER  COMPLETEFAILOVERCLUSTER | * Applies to the /FAILOVERCLUSTERGROUP setting * An existing group that already contains clustered service(s) was attempted to be used for the failover cluster * Remove any network name, IP address, or services from the group * Use a different group that does not contain existing resources, or let Setup create a new group instead |
| IP address not found. To create a SQL Server failover cluster, specify at least one IP address. | INSTALLFAILOVERCLUSTER  COMPLETEFAILOVERCLUSTER | * Applies to the /FAILOVERCLUSTERIPADDRESSES setting * The setting value is empty * Refer to the detail log to determine why Setup did not generate any default IP addresses for the cluster networks * Confirm that the roles for the cluster networks are set correctly to allow client access where appropriate * Confirm that the cluster networks are up |
| The IP Address '%1' is invalid. | INSTALLFAILOVERCLUSTER  COMPLETEFAILOVERCLUSTER | * Applies to the /FAILOVERCLUSTERIPADDRESSES setting * The setting value is missing the IP address * The type of the address does not match the type specified in the setting (for example, “IPv6” was specified but the address is IPv4) * For IPv4 addresses, the address is not correct (that is, it doesn’t match the format XXX.XXX.XXX.XXX) * For IPv4 addresses, a portion of the address is > 255 (for example, 346.0.0.1) |
| The specified IP Address '%1' is a reserved IP address. To continue, specify an IP address that does not end in '.0'. | INSTALLFAILOVERCLUSTER  COMPLETEFAILOVERCLUSTER | * Applies to the /FAILOVERCLUSTERIPADDRESSES setting * For IPv4 addresses, confirm that the last quartet of the address is not .0 |
| The IP Address '%1' is already in use. Pick another IP address to continue. | INSTALLFAILOVERCLUSTER  COMPLETEFAILOVERCLUSTER | * Applies to the /FAILOVERCLUSTERIPADDRESSES setting * For IPv4 addresses, the address is assigned to a device on the network |
| The subnet mask '%1' is invalid. To continue, specify a valid subnet mask, then try again. | INSTALLFAILOVERCLUSTER  COMPLETEFAILOVERCLUSTER | * Applies to the /FAILOVERCLUSTERIPADDRESSES setting * The subnet mask was not specified in the setting * For IPv4 addresses, the mask is not correct (that is, it doesn’t match the format XXX.XXX.XXX.XXX) * For IPv4 addresses, a portion of the mask is > 255 (for example, 346.0.0.1) |
| IPv6 is not supported for clustering on OS versions lower than Windows Server 2008. | INSTALLFAILOVERCLUSTER  COMPLETEFAILOVERCLUSTER | * Applies to the /FAILOVERCLUSTERIPADDRESSES setting * IPv6 is only supported on Windows Server 2008 operating systems |
| No cluster networks have a bound IPv6 address which indicates IPv6 is not supported on this machine. Bind an address to the cluster name or other resource to continue. | INSTALLFAILOVERCLUSTER  COMPLETEFAILOVERCLUSTER | * Applies to the /FAILOVERCLUSTERIPADDRESSES setting * Verify that at least one of the cluster networks available has at least one active IPv6 address bound to it |
| The IP Address '%1' cannot use DHCP because it is not supported on this operating system. | INSTALLFAILOVERCLUSTER  COMPLETEFAILOVERCLUSTER | * Applies to the /FAILOVERCLUSTERIPADDRESSES setting * DHCP is only supported for SQL Server 2008 instances installed on Windows Server 2008 operating systems |
| The IP Address '%1' has an invalid network name '%2'. To continue, specify an IP address with a valid network name and retry. | INSTALLFAILOVERCLUSTER  COMPLETEFAILOVERCLUSTER | * Applies to the /FAILOVERCLUSTERIPADDRESSES setting * The network name portion of the setting does not specify a valid network on the cluster |
| The cluster network '%1' does not have a bound IPv6 address which indicates IPv6 is not supported for that network. Bind an address to the cluster name or other resource to continue. | INSTALLFAILOVERCLUSTER  COMPLETEFAILOVERCLUSTER | * Applies to the /FAILOVERCLUSTERIPADDRESSES setting * Verify that the cluster network specified for the IP address has at least one active IPv6 address bound to it |
| No IP Address cluster resources could be found for the SQL Server failover cluster instance name '%1'. | UPGRADE  REPAIR  REBUILDDATABASE | * Applies to the /FAILOVERCLUSTERIPADDRESSES setting * This validation only runs if the group the instance is installed into was found * The error is reported when the group the instance is installed in has no IP address resources * Manually repair the group to restore the missing resources, and run the scenario again |
| The instance specified is either not installed, or not prepared for use with failover clustering. Confirm that the correct SQL Server instance name was specified, and then run Setup again. | INSTALLFAILOVERCLUSTER  COMPLETEFAILOVERCLUSTER | * Applies to the internal /FAILOVERCLUSTERNODES setting * The list of nodes is automatically calculated * Verify that the instance name specified by the user is installed on some nodes of the cluster |
| The cluster node name '%1' is invalid. The specified computer is not a cluster node in this SQL Server failover cluster. | INSTALLFAILOVERCLUSTER  COMPLETEFAILOVERCLUSTER | * Applies to the internal /FAILOVERCLUSTERNODES setting * The list of nodes is automatically calculated * Verify that all the nodes reported by Setup in the instance are active members of the Windows cluster |
| A clustered instance of SQL Server could not be found to upgrade. Possible reasons are that the wrong instance name was specified, the instance is not clustered, or Setup could not determine the failover cluster network name based on the instance name. Confirm that the correct information was specified, and then run Setup again. | UPGRADE | * Applies to the internal /FAILOVERCLUSTERNODES setting * The list of nodes is automatically calculated * Verify that the instance name specified by the user is installed on some nodes of the cluster |
| Based on detected settings, the SQL Server instance is clustered, but no clustered features could be found. SQL Server Setup is unable to repair the instance due to the mismatch in detected settings. To continue, confirm that the failover cluster network name is correct and that the installed features were successfully clustered. | REPAIR  REBUILDDATABASE | * Applies to the internal /FAILOVERCLUSTERNODES setting * The list of nodes is automatically calculated * Verify that the instance name specified by the user is installed on some nodes of the cluster |
| The local computer is the only node left in the failover cluster that has not been upgraded, or it is a single-node failover cluster. As part of the upgrade of the local computer, ownership of the failover cluster group must be transferred to the upgraded version of SQL Server. The only valid options for the %1 setting in this scenario are 1 or 2. | UPGRADE | * Applies to the /FAILOVERCLUSTERROLLOWNERSHIP setting * Remove the setting from the command line, or change the value to 1 or 2 |
| The failover cluster group is already owned by the upgraded nodes of the SQL Server instance, so it is not possible to transfer ownership to the newer version again. The only valid options for the %1 setting in this scenario are 0 or 2. | UPGRADE | * Applies to the /FAILOVERCLUSTERROLLOWNERSHIP setting * Remove the setting from the command line, or change the value to 0 or 2 |
| The local node of the failover cluster has already been upgraded, and no other nodes have been upgraded. It is not possible to run upgrade on the local node again. | UPGRADE | * Applies to the /FAILOVERCLUSTERROLLOWNERSHIP setting * Remove the setting from the command line, or change the value to 0 or 2 * Run upgrade on a another node of the cluster, because the local node was already upgraded |
| None of the nodes in the failover cluster have been upgraded yet. Because of this, it is not possible for the ownership of the failover cluster group to be transferred to the upgraded version of SQL Server. The only valid options for the %1 setting in this scenario are 0 or 2. | UPGRADE | * Applies to the /FAILOVERCLUSTERROLLOWNERSHIP setting * Remove the setting from the command line, or change the value to 0 or 2 |
| The SQL Server failover cluster instance '%1' was not correctly detected. The instance was discovered on the local node but it was not found to be active. To continue, confirm the state of the instance installed on all applicable nodes of the cluster and the state of the failover cluster resources. | UPGRADE  REPAIR  REBUILDDATABASE | * Applies to the internal /ISLOCALNODEACTIVE setting * The error occurs when the local node is not the owner of the cluster group, but Setup was unable to detect any other nodes in the cluster * Verify that the registry of the remote machines is accessible to the local node * The list of nodes is automatically calculated * Verify that the instance name specified by the user is installed on some nodes of the cluster |
| The specified network name for this SQL Server failover cluster instance is invalid. The name cannot be null or empty. | INSTALLFAILOVERCLUSTER  COMPLETEFAILOVERCLUSTER  ADDNODE  UPGRADE  REPAIR  REBUILDDATABASE | * Applies to the /FAILOVERCLUSTERNETWORKNAME setting * The setting must be specified for the INSTALLFAILOVERCLUSTER or COMPLETEFAILOVERCLUSTER scenarios * For the scenarios where the instance is already installed, it means the Setup was unable to determine the network name * For the scenarios where the instance is already installed, verify the “<InstId>\Cluster\ClusterName” registry value exists on all nodes of the cluster |
| The specified network name for this SQL Server failover cluster instance is too long. Network names are limited to %1 characters. | INSTALLFAILOVERCLUSTER  COMPLETEFAILOVERCLUSTER  ADDNODE  UPGRADE  REPAIR  REBUILDDATABASE | * Applies to the /FAILOVERCLUSTERNETWORKNAME setting * The name specified by the user is too long |
| The SQL Server failover cluster instance name '%1' could not be found as a cluster resource. | ADDNODE  UPGRADE  REPAIR REBUILDDATABASE | * Applies to the calculated value of the /FAILOVERCLUSTERNETWORKNAME setting * The resource for the network name was not found in the cluster * Verify that the <InstId>\Cluster\ClusterName registry value is correct |
| The SQL Server failover cluster instance name '%1' already exists as a clustered resource. Specify a different failover cluster instance name. | INSTALLFAILOVERCLUSTER  COMPLETEFAILOVERCLUSTER | * Applies to the calculated value of the /FAILOVERCLUSTERNETWORKNAME setting * A network name resource was found in the cluster whose Name private property is the same as the current value specified in the setting * Remove the existing network name resource and run Setup again with a different failover cluster network name value |
| The SQL Server failover cluster network name '%1' is not valid. The name must be a valid DNS and WinS name. | INSTALLFAILOVERCLUSTER  COMPLETEFAILOVERCLUSTER | * Applies to the calculated value of the /FAILOVERCLUSTERNETWORKNAME setting * The value for the setting is not a valid Domain Name System (DNS) host name label * Retry Setup after specifying a different failover cluster network name |
| The SQL Server failover cluster instance name '%1' already exists as a server on the network. Specify a different failover cluster instance name. | INSTALLFAILOVERCLUSTER  COMPLETEFAILOVERCLUSTER | * Applies to the calculated value of the /FAILOVERCLUSTERNETWORKNAME setting * A server on the network already exists with the specified network name * Retry Setup after specifying a different failover cluster network name |
| The given network name is unusable because there was a failure trying to determine if the network name is valid for use by the clustered SQL instance due to the following error: '%1' | INSTALLFAILOVERCLUSTER  COMPLETEFAILOVERCLUSTER | * Applies to the calculated value of the /FAILOVERCLUSTERNETWORKNAME setting * This error means an error was returned by a call to the Win32® NetServerGetInfo() API * Refer to the detail log for the reasons for the failure to determine if the network name is already in use * Verify connectivity to the network / Active Directory Domain Services (AD DS) |
| The volume containing SQL Server data directory %1 does not belong to the cluster group. | INSTALLFAILOVERCLUSTER  COMPLETEFAILOVERCLUSTER | * Applies to the /INSTALLSQLDATADIR and related data directory settings * Make sure that the path specified is on one of the disks selected for the cluster group (SLP setting /FAILOVERCLUSTERDISKS) |
| Missing domain group. To continue, provide a valid domain group for the Database Engine service. | PREPAREFAILOVERCLUSTER  INSTALLFAILOVERCLUSTER  UPGRADE  REPAIR | * Applies to the/SQLDOMAINGROUP setting * Required only on Windows Server 2003 for PREPAREFAILOVERCLUSTER and INSTALLFAILOVERCLUSTER, and UPGRADE from SQL Server 2000. May be required for REPAIR if the domain group cannot be found. |
| Invalid domain group. To continue, provide a valid domain group for the Database Engine service. | PREPAREFAILOVERCLUSTER  INSTALLFAILOVERCLUSTER  UPGRADE  REPAIR | * Applies to the /SQLDOMAINGROUP setting * Verify that the group account exists and is in the same domain forest as the domain to which the cluster node is joined * For Windows 2000 Server mixed-mode domains, this cannot be a domain local group. However, this situation is not diagnosed by Setup. It will show up at service startup with access-denied errors on the SQL Server error log. |
| The service account is not valid. To continue, provide a domain account for Database Engine service. | PREPAREFAILOVERCLUSTER  INSTALLFAILOVERCLUSTER  REPAIR | * Applies to the /SQLSVCACCOUNT setting * Verify that the service account specified exists and is in the same domain forest as the domain to which the cluster node is joined |
| The service account is not in the domain group. Ensure the service account is a member of the domain group for the Database Engine. | PREPAREFAILOVERCLUSTER  INSTALLFAILOVERCLUSTER  REPAIR | * Applies to the /SQLSVCACCOUNT setting * Verify that the service account is in the domain group specified by the /SQLDOMAINGROUP setting |
| The SQL Server service domain group or account does not match with that on the remaining cluster nodes. | COMPLETEFAILOVERCLUSTER  REPAIR | * Indicates that PREPAREFAILOVERCLUSTER was run for the same instance id on different cluster nodes without specifying identical values for /SQLDOMAINGROUP and /SQLSVCACCOUNT * You must run REMOVENODE on each of the nodes that are incorrect and then rerun PREPAREFAILOVERCLUSTER specifying the correct values |
| Invalid or missing username or password. To continue, provide valid credentials for the Database Engine service. | PREPAREFAILOVERCLUSTER  INSTALLFAILOVERCLUSTER  ADDNODE  REPAIR | * This applies to the /SQLSVCACCOUNT and /SQLSVCPASSWORD settings * In particular, for ADDNODE, the password needs to be respecified for the Database Engine service because the password cannot be acquired from the active node |
| SQLBrowser: SQL Server Browser Install for feature 'SQL\_Browser\_Redist\_SqlBrowser\_Cpu32' generated exception, and will invoke retry option. The exception: Microsoft.SqlServer.Configuration.SqlBrowser.SqlBrowserConfigException: Failed to open registry key 'SOFTWARE\Microsoft\Microsoft SQL Server\90' in order to apply permissions on it. The registry key does not exist. | PREPAREFAILOVERCLUSTER  INSTALLFAILOVERCLUSTER  ADDNODE  (Note: SQL Server Browser does not have cluster-specific failures. This error also applies to other Setup scenarios) | * SQL Server Browser does not have cluster specific failures. This error indicates that the Database Engine or Analysis Services MSI is not installed correctly; however, this is not cluster-specific. |
| Must provide AGTDOMAINGROUP parameter | INSTALLFAILOVERCLUSTER  PREPAREFAILOVERCLUSTER  REPAIR  UPGRADE | * The AGTDOMAINGROUP parameter is missing or empty. * In repair and upgrade scenarios, this message appears only for clustered instances. * This input setting is required only for upgrade from a SQL Server 2000 cluster. For other upgrades, a domain group is already associated with instance that is upgraded, so this input setting is not necessary. * In repair scenarios on clustered instances, this input setting is required only if the cache of domain group is missing from registry. |
| Parameter AGTDOMAINGROUP is not a valid domain group | INSTALLFAILOVERCLUSTER  PREPAREFAILOVERCLUSTER  REPAIR  UPGRADE | * Use a domain group and not a local machine group as the SQL Service Agent service domain group (for example, redmond\sqlsetup-agtgroup). |
| Local Service is not a supported account for SQL Server Agent. | INSTALLFAILOVERCLUSTER  PREPAREFAILOVERCLUSTER  INSTALL  REPAIR  ADDNODE | * Use another account besides LOCAL SERVICE for the SQL Server Agent service account when installing a clustered or standalone instance. |
| The credentials you provided for the SQL Server Agent service are invalid. To continue, provide a valid account and password for the SQL Server Agent service. | INSTALLFAILOVERCLUSTER  PREPAREFAILOVERCLUSTER  INSTALL  REPAIR  ADDNODE | * The SQL Server Agent service account and password do not match or were not provided. |
| The specified service account is not valid. To continue, provide a domain account for SQL Server Agent service. | INSTALLFAILOVERCLUSTER  PREPAREFAILOVERCLUSTER  REPAIR (Cluster only)  ADDNODE | * Use a domain account as the SQL Server Agent service account when installing a clustered instance (for example, redmond\sqlcl01) |
| The service account you specify for the SQL Server Agent service must be a member of the SQL Server Agent domain group. | INSTALLFAILOVERCLUSTER  PREPAREFAILOVERCLUSTER  COMPLETEFAILOVERCLUSTER | * Validate that that the SQL Server Agent service account is part of the SQL Server Agent service domain group for clustered scenarios. |
| The SQL Agent service account name and SID must be the same across all cluster nodes. | COMPLETEFAILOVERCLUSTER  ADDNODE | * When running the enterprise installation scenario (that is, prepare on n-nodes and then run complete on one of them), ensure that the SQL Server Agent service accounts are the same on all prepared nodes for the particular instance. * On AddNode, the SQL Server Agent service account specified must be same as it is on the other nodes of the SQL Server cluster. |
| The SQL Agent domain group name and SID must be the same across all cluster nodes. | COMPLETEFAILOVERCLUSTER | * When running the enterprise installation scenario (that is, prepare on n-nodes and then run complete on one of them), ensure that the SQL Server Agent service domain groups are the same on all prepared nodes for the particular instance. |
| The SQL Agent service account cannot be NETWORK SERVICE and a member of the SQL Agent domain group. | INSTALLFAILOVERCLUSTER  PREPAREFAILOVERCLUSTER  REPAIR (Cluster only)  ADDNODE | * Use an account other than NETWORK SERVICE for the SQL Server Agent service account when installing a clustered instance. |
| The specified SQL Server instance does not have a SQL Agent service that Setup can successfully start. | UPGRADE (Standalone instance only) | * The SQL Server Agent service of the SQL Server instance to be upgraded cannot be started. Ensure that the SQL Server Agent service can start before performing the upgrade. * The associated check is performed for an upgrade of a stand-alone instance only. |
| On a domain controller, the SQL Server Agent service account cannot be a built-in account. | INSTALL  REPAIR | * For SQL Server instances on a domain controller machine, the SQL Server Agent service account cannot be a built-in account such as Network Service. Use a user account instead. |
| The AGTSVCSTARTUPTYPE start mode must be Manual for Clustering. | INSTALLFAILOVERCLUSTER  PREPAREFAILOVERCLUSTER | * The service startup mode must be Manual when installing a clustered instance. |
| The specified SQL Server Agent service account is not a valid user account. | INSTALLFAILOVERCLUSTER  PREPAREFAILOVERCLUSTER  INSTALL  REPAIR  ADDNODE | * The SQL Server Agent service account has to be a user account. It cannot be a local group such as BUILTIN\Administrators. |
| The SQL Agent service domain group could not be recovered. To complete repair, SQL Agent Service domain group must be provided. | REPAIR (Cluster only) | * In repairing the SQL Server Agent service of a clustered instance, the domain group of SQL Server Agent could not be found. To complete repair, provide the SQL Server Agent domain group that was specified during installation of the instance being repaired. |
| The SQL Server Agent service (SQLServerAgent) is missing and must be re-created as part of the repair operation. To continue, run SQL Server repair at a command prompt. Include the user name and password credentials for the SQL Server Agent service account. Note that the SQL Server Agent user name and password credentials that you provide must match the credentials for SQL Server Agent on all other SQL Server failover cluster nodes in the instance of SQL Server . | REPAIR(Cluster only) | * In repair of a clustered SQL Server instance, the SQL Server Agent service is missing. To re-create the service, the original SQL Server Agent service account needs to be provided. Note that in nonclustered instance, Setup uses NETWORK SERVICE as default service account, instead of throwing this message, but in cluster scenarios, there is no such option. |
| The specified SQL Server instance does not have a valid SQL Server Agent service. | UPGRADE  REBUILDDATABASE | * While performing Upgrade or RebuildDatabase Setup action, the SQL Server Agent service is found to be missing. To continue, repair the SQL Server instance so that the SQL Server Agent service is restored. |
| The SQL Server Agent service (SQLServerAgent) is missing and must be re-created as part of the repair operation. Since the user name and password credentials for the SQL Server Agent service account was not provided, the service account will be set to NT AUTHORITY\\NetworkService and the service will be disabled. To enable the service or change the service account, use the SQL Server Configuration Manager. | REPAIR | * Self-explanatory. Intended to inform user that the service is disabled and ask them to change service account using SQL Server Configuration Manager. |
| The SQL Server repair operation has set registry keys to default values, as follows: '%1', field: '%2'. Previous value: '%2'. Current (default) value: '%3'. | REPAIR | * In repair, the values in the registry key for SQLServerAgent (HKLM\Software\Microsoft\Microsoft SQL Server\<InstanceID>\SQLServerAgent) are set to their default values. This message is displayed in summary for each value that was nondefault and set to default, so that user knows which registry values have changed as part of the repair. These are the same registry key values that are populated during a fresh installation. |
| The SQL Server removal of SQL Server failover cluster '%1' encountered an error deleting cluster resource '%2'. The exception message is '%2'. The operation is not blocked. After SQL Server Setup completes, use the Windows cluster admin tool to manually delete the cluster resource. | REMOVENODE | * Setup did not successfully remove the SQL Server Agent cluster resource of the instance being removed. This message informs the user about manually performing the clean-up. |
| The Analysis Services service account is not a valid domain account. | PREPAREFAILOVERCLUSTER INSTALLFAILOVERCLUSTER | * This error happens if the user has provided an invalid username that is not a valid domain account. Verify the user name and check whether it is a domain account and continue the process. |
| The Analysis Services domain group does not exist. To continue, specify a domain group for Analysis Services, then try again. | PREPAREFAILOVERCLUSTER INSTALLFAILOVERCLUSTER | * This error happens if the domain group provided is invalid. Provide a valid domain group and continue the process. |
| 1) Please select a valid Shared Disk.  2) The folder path specified is invalid. Please enter a valid folder path.  3) Unable to access the specified folder. Please enter a valid folder path. | COMPLETEFAILOVERCLUSTER  INSTALLFAILOVERCLUSTER | * These error occurs when the given folder path is not shared, not valid, or not accessible. Verify the folder path and continue the process. This error is applicable to all folders such as data, log, temp, backup, and config. |
| Service account doesn't match on remote nodes. | COMPLETEFAILOVERCLUSTER | * This error happens when the user name provided is different across prepared nodes. Check the user name under which the service is running across all nodes and continue the process. You can also use the detail log to find out exactly which node is out of sync. |
| Domain group doesn't match on remote nodes. | COMPLETEFAILOVERCLUSTER | * Steps are same as for the previous message, except in this case domain group name doesn’t match. This error occurs only with Windows Server 2003. |
| 1) You must provision the system with at least one system administrator.  2) User or group does not exist. | COMPLETEFAILOVERCLUSTER  INSTALLFAILOVERCLUSTER | * This error happens when the user didn't provide the sys admin account or provided invalid sys admin account. Provide the valid sys admin account and proceed. |
| Unable to retrieve Service Account from the Active Node. | ADDNODE | * This error happens when the Setup is unable to reach the active node. Verify whether the user can reach the active node from the machine they are trying add node. There might be a networking connectivity issue. |
| User Name doesn't match active node account. | ADDNODE | * The service account doesn't match on the active node. Verify whether they are the same and continue the process. |
| Error 1603 (setup error) | REMOVENODE | * This happens if the cluster registry hive is missing some keys. * The workaround for this failure is to set the Setup\SqlCluster value to 0 and start the removal node process. |
| The Analysis Services service could not be started due to a logon failure. Use the SQL Configuration Manager to set the username and password on the active nodes and on all of the passive nodes. | REPAIR | * This happens if the user and password are not correct. Use SQL Server Configuration Manager to correct the password on all nodes. |
| The failover instance name could not be retrieved. The settings cannot be updated. Check whether the Active Node is accessible from the given node. | UPGRADE | * During an upgrade, the active node must be online. If not it will cause the upgrade process to fail. Bring up the active node before you start the upgrade. |
| Unable to connect to Analysis Services server, '%1', to update settings. Change the settings manually at the end of the upgrade. | UPGRADE | * At the end of the upgrade process, Analysis Services connects to provisioning. If the provisioning could not be done, an error occurs. Manually add the OLAP administrator after the upgrade. |
| Unable to connect to Analysis Services server, '%1'. Ensure that the service is accessible. | UPGRADE | * Make sure the firewall is not blocking access to the active node during the upgrade. |
| The user must be an OLAP administrator to update the settings. Change the settings manually at the end of the upgrade. | UPGRADE | * The account under which Setup runs must be an OLAP administrator in order to update the config values. If you receive this error, the account does not have OLAP administrator permissions; after the upgrade, you can manually change the settings. |
| The cluster resource is not online. Bring the Analysis Services server online before starting the upgrade process. | UPGRADE | * The resource needs to be online during the upgrade. |
| Service Account, '%1', is not part of the domain group, '%2'. | REPAIR | * To solve this problem, add the service account to the domain group |
| The service account could not be retrieved. Use the ASSVCACCOUNT service account command line parameter to specify the service account. | REPAIR | * During repair, Setup was unable to get the service account. Pass the service account and password in the command line to proceed with the repair process. |
| The configuration path could not be retrieved from the active node. Check whether you can reach the active node from the given node. | REPAIR | * During repair, Setup tries retrieve the config path from active node; if Setup is unable to retrieve the information, Setup was not able reach the active node. Ensure that the active node can be reached from the computer you are repairing. |
| No user name was provided. Use the ASSVCACCOUNT command line parameter to specify a user name. | REPAIR | * During repair, Setup was unable to get the service account. Pass the service account and password in the command line to proceed with the repair process. |
| Unable to connect to the active node. | REPAIR | * During repair, make sure the passive node can reach the active node. Verify that there are no network problems. |
| The local service account is different from the active node. Use the SQL Configuration Manager to change the service account to match the active node. | REPAIR | * During repair, Setup validates whether the passive node service account is the same as active node service account. If it is different, change the account using SQL Server Configuration Manager. |
| The domain group, '%1', does not exist on the active node. Specify a valid domain group. | REPAIR | * The domain group might be missing. Provide the domain group through a command line parameter for repair. |
| The domain group does not exist on the active node. Specify a valid domain group. | REPAIR | * The domain group might be missing; provide the domain group through a command line parameter for repair. |
| The Service Start Mode parameter cannot be disabled on the active node. Change the service state to either manual or automatic using the SQL Configuration Manager. | REPAIR | * None. |
| The domain group was not provided. Specify the domain group on the command line. | REPAIR | * The domain group might be missing; provide the domain group through a command line parameter for repair. |
| The configuration path was not provided. Provide the configuration path on the command line. | REPAIR | * The configuration path might be missing on the active node; provide a configuration path through command line |
| The configuration path is not valid. Specify a valid configuration path. | REPAIR | * Provide valid configuration path through the command line |
| The configuration path is not valid on the active node. Specify a valid configuration path on the active node. | REPAIR | * Make sure the configuration path is valid and accessible |
| The configuration path is not accessible. Check whether you can access the disk from the given machine. | REPAIR | * Make sure the configuration path is valid and accessible |
| The configuration path is not part of a valid shared disk. Specify a valid shared disk. | REPAIR | * Make sure the configuration path is valid and accessible |

## Troubleshooting Error Messages in “\*ErrorLog” Files

| Error message | Scenarios | Troubleshooting tips |
| --- | --- | --- |
| Removal of clustered feature %1 is not supported | UNINSTALL | * This happens when the user tries to remove a clustered feature via /ACTION UNINSTALL instead of /ACTION REMOVENODE * The value of 1 in the <InstId>\ClusterState key triggers this block |
| Install of feature %1 is not supported on clustered instance | INSTALL  UPGRADE | * This occurs when the instance the user specified is already installed on the local machine, and the instance contains clustered features, and the user is trying to add additional stand-alone features. It is not possible to install nonclustered features to a clustered instance. * Values of 1 or 2 in the <InstId>\ClusterState key trigger this block |
| The CPU platform for the %1 feature is inconsistent between the nodes of the cluster. To cluster an instance, the platform of the instance on each node must be the same. | COMPLETEFAILOVERCLUSTER  ADDNODE | * This occurs when Setup believes the platform of the instance is not consistent on some of the nodes, for example, one instance is 32-bit and the other one is 64-bit * For remote detection, the platform of the installation is determined by which side of the registry the <InstId>\Setup\ProductCode registry value was found; if it was found in the 32-bit registry, the instance is considered 32-bit |
| The installed language for the %1 feature is inconsistent between the nodes of the cluster. To cluster an instance, the language of the instance on each node must be the same. | COMPLETEFAILOVERCLUSTER  ADDNODE | * This occurs when Setup believes the language of the instance is not consistent on some of the nodes, for example, one instance is English and the other is Japanese * For remote detection, the language of the installation is determined by reading the <InstId>\Setup\Language value in the registry |
| The installation directory for the %1 feature is inconsistent between the nodes of the cluster. To cluster an instance, each instance must be installed to the same path on each machine. | COMPLETEFAILOVERCLUSTER | * This occurs when the installation path for the instance is not consistent on some of the nodes, for example, one instance is installed to C:\SQL and another is installed to D:\SQL * The installation path is determined from discovery, which looks at the <InstId>\Setup\SQLPath value in the registry |
| The SQL Server feature '%1' is not in a supported state for repair, as it was installed in preparation to be clustered. Cluster prepared features cannot be repaired. To continue, uninstall the specified SQL Server feature. | REPAIR | * This occurs if the customer tries to run Repair against a cluster prepared instance * If the prepared instance failed during configuration, uninstall it and run PrepareFailoverCluster again |

# Conclusion

SQL Server is at the core of many mission critical applications. Data protection and high availability is core to providing mission critical services. SQL Server 2008 failover clustering is one of the primary technologies that enable businesses to achieve high availability with the SQL Server product. SQL Server 2008 builds upon previous versions and integrates with other SQL Server features, which enables implementations of SQL Server to achieve even greater levels of availability in production environments. When this technology is combined with the people and process aspects of achieving high availability, it enables businesses and users to achieve their availability goals.

For more information, see the following Web sites:

* The SQL Server Web site at <http://www.microsoft.com/sqlserver/>
* The SQL Server TechCenter Web site at <http://technet.microsoft.com/en-us/sqlserver/>
* The SQL Server DevCenter Web site at <http://msdn.microsoft.com/en-us/sqlserver/>

Did this paper help you? Please give us your feedback. Tell us on a scale of 1 (poor) to 5 (excellent), how would you rate this paper and why have you given it this rating? For example:

Are you rating it high due to having good examples, excellent screen shots, clear writing, or another reason?

Are you rating it low due to poor examples, fuzzy screen shots, or unclear writing?

This feedback will help us improve the quality of white papers we release.

[Send feedback](mailto:sqlfback@microsoft.com?subject=White%20Paper%20Feedback:%20SQL%20Server%202008%20Failover%20Clustering).