**Problem**After Installing [SQL Server 2008 Installation Process](http://mssqltips.com/tip.asp?tip=1678) you must have noticed how different SQL Server 2008 installation is from its previous versions. Now, we will be installing SQL Server 2008 on a Windows Server 2008 Cluster. Windows Server 2008 has a lot of differences from its previous versions and one of them is the clustering feature.

**Solution**There have been a lot of changes regarding clustering between Windows Server 2003 and Windows Server 2008. It took quite a lot of effort for us to build a cluster in Windows Server 2003 – from making sure that the server hardware for all nodes are cluster-compatible to creating resource groups. Microsoft has redefined clustering with Windows Server 2008, making it simpler and easier to implement. Now that both SQL Server 2008 and Windows Server 2008 are out in the market for quite some time, it would be a must to prepare ourselves to be able to setup and deploy a clustered environment running both. Installing SQL Server on a stand-alone server or member server in the domain is pretty straight-forward. Dealing with clustering is a totally different story. The goal of this series of tips is to be able to help DBAs who may be charged with installing SQL Server on a Windows Server 2008 cluster.

***Prepare the cluster nodes***

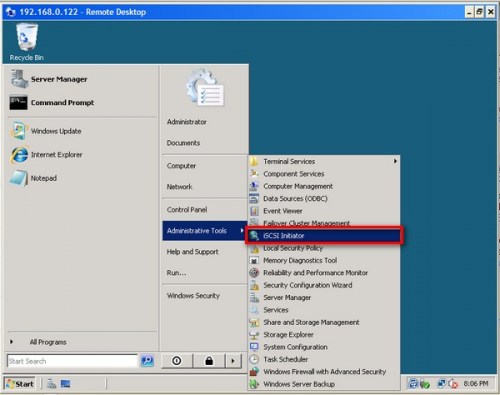
I will be working on a 2-node cluster throughout the series and you can extend it by adding nodes later on. You can do these steps on a physical hardware or a virtual environment. I opted to do this on a virtual environment running VMWare. To start with, download and install a copy of the evaluation version of [Windows Server 2008 Enterprise Edition.](http://www.microsoft.com/downloads/details.aspx?FamilyId=13C7300E-935C-415A-A79C-538E933D5424&displaylang=en) This is pretty straight-forward and does not even require any product key or activation. Evaluation period runs for 60 days and can be extended up to 240 days so you have more than enough time to play around with it. Just make sure that you select at least the Enterprise Edition during the installation process and have at least 12GB of disk space for your local disks. This is to make sure you have enough space for both Windows Server 2008 and the binaries for SQL Server 2008. A key thing to note here is that you should already have a domain on which to join these servers and that both have at least 2 network cards – one for the public network and the other for the heartbeat. Although you can run a cluster with a single network card, it isn’t recommend at all. I’ll lay out the details of the network configuration as we go along. After the installation, my recommendation is to immediately install [.NET Framework 3.5 with Service Pack 1](http://www.microsoft.com/downloads/details.aspx?familyid=ab99342f-5d1a-413d-8319-81da479ab0d7&displaylang=en) and [Windows Installer 4.5](http://www.microsoft.com/downloadS/details.aspx?familyid=5A58B56F-60B6-4412-95B9-54D056D6F9F4&displaylang=en) (the one for Windows Server 2008 x86 is named ***Windows6.0-KB942288-v2-x86.msu***). These two are prerequisites for SQL Server 2008 and would speed up the installation process later on.

***Carve out your shared disks***

We had a lot of challenges in Windows Server 2003 when it comes to shared disks that we will use for our clusters. For one, the 2TB limit which has a lot to do with the master boot record (MBR) has been overcome by having the GUID Partition Table (GPT) support in Windows Server 2008. This allows you to have 16 Exabytes for a partition. Another has been the use of directly attached SCSI storage. This is no longer supported for Failover Clustering in Windows Server 2008. The only supported ones will be Serially Attached Storage (SAS), Fiber Channel and iSCSI. For this example, we will be using an iSCSI storage with the help of an iSCSI Software Initiator to connect to a software-based target. I am using [StarWind’s iSCSI SAN](http://www.starwindsoftware.com/starwind-server-iscsi-san) to emulate a disk image that my cluster will use as shared disks. In preparation for running SQL Server 2008 on this cluster, I recommend creating at least 4 disks – one for the quorum disk, one for MSDTC, one for the SQL Server system databases and one for the user databases. Your quorum and MSDTC disks can be as small as 1GB, although [Microsoft TechNet](http://technet.microsoft.com/en-us/library/cc770620.aspx) specifies a 512MB minimum for the quorum disk. If you decide to use iSCSI as your shared storage in a production environment, a dedicated network should be used so as to isolate it from all other network traffic. This also means having a dedicated network card on your cluster nodes to access the iSCSI storage.

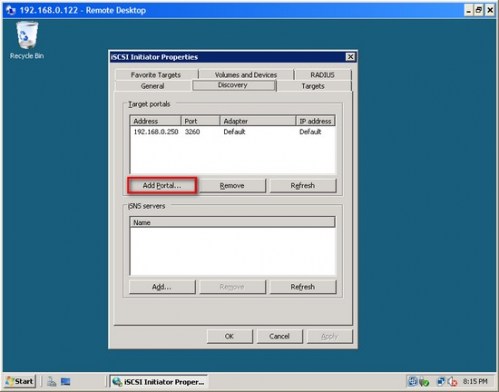
***Present your shared disks to the cluster nodes***

Windows Server 2008 comes with iSCSI Initiator software that enables connection of a Windows host to an external iSCSI storage array using network adapters. This differs from previous versions of Microsoft Windows where you need to [download](http://www.microsoft.com/downloads/details.aspx?familyid=12cb3c1a-15d6-4585-b385-befd1319f825&displaylang=en) and install this software prior to connecting to an iSCSI storage. You can launch the tool from **Administrative Tools** and select **iSCSI Initiator**.

[](http://dbperf.files.wordpress.com/2010/07/1687_1.jpg)

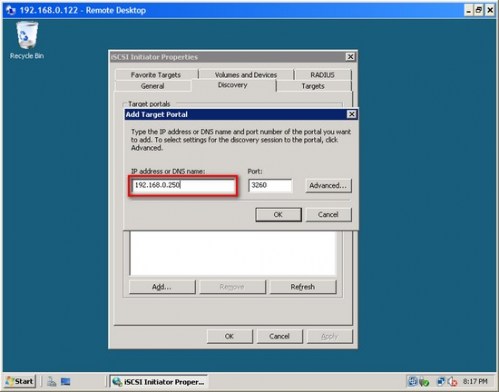
To connect to the iSCSI target:

In the **iSCSI Initiator Properties** page, click on the **Discovery** tab.

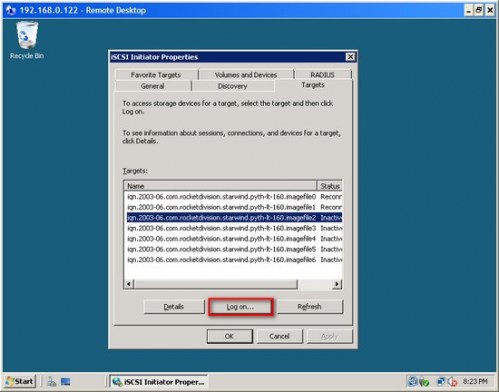
[](http://dbperf.files.wordpress.com/2010/07/1687_21.jpg)

Under the **Target Portals** section, click on the **Add Portal** button.

In the **Add Target Portal** dialog, enter the DNS name or IP address of your iSCSI Target and click **OK**. If you are hosting the target on another Windows host as an image file, make sure that you have your Windows Firewall configured to enable inbound traffic to port 3260. Otherwise, this should be okay.

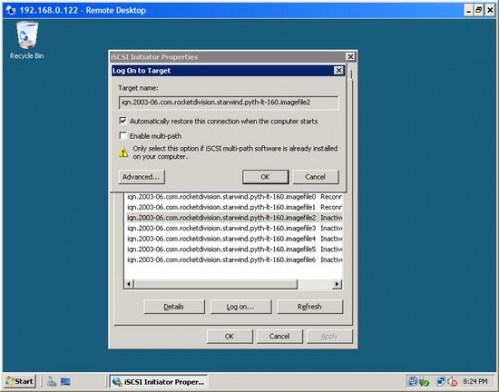
[](http://dbperf.files.wordpress.com/2010/07/1687_3.jpg)

Back in the **iSCSI Initiator Properties** page, click on the **Targets** tab. You should see a list of the iSCSI Targets that we have defined earlier

[](http://dbperf.files.wordpress.com/2010/07/1687_4.jpg)

Select one of the targets and click on the **Log on** button.

In the **Log On to Target** dialog, select the **Automatically restore this connection when the computer starts** checkbox. Click **OK**.

[](http://dbperf.files.wordpress.com/2010/07/1687_5.jpg)

Once you are done, you should see the status of the target change to **Connected**. Repeat this process for all the target disks we initially created on both of the servers that will become nodes of your cluster.

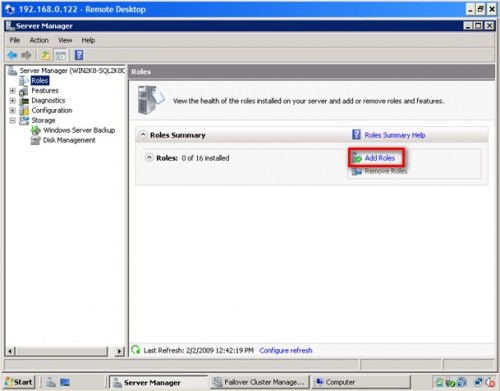
Once the targets have been defined using the iSCSI Initiator tool, you can now bring the disks online, initialize them, and create new volumes using the Server Manager console. I won’t go into much detail on this process as it is similar to how we used to do it in Windows Server 2003, except for the new management console. After the disks have been initialized and volumes created, you can try logging in to the other server and verify that you can see the disks there as well. You can rescan the disks if they haven’t yet appeared.

***Adding Windows Server 2008 Application Server Role***

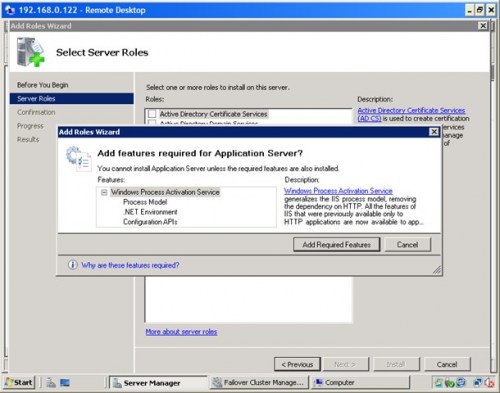
Since we will be installing SQL Server 2008 later on, we will have to add the Application Server role on both of the nodes. A server role is a program that allows Windows Server 2008 to perform a specific function for multiple clients within a network. To add the Application Server role,

Open the **Server Manager** console and select **Roles**.

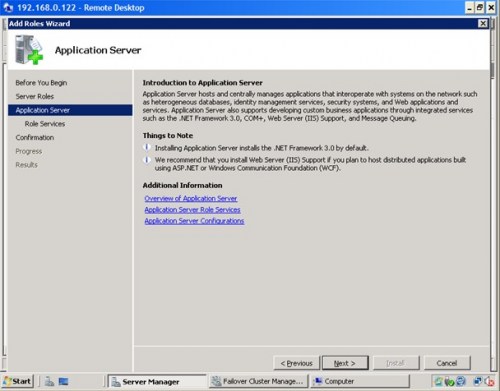
Click the **Add Roles** link. This will run the **Add Roles Wizard**

[](http://dbperf.files.wordpress.com/2010/07/1687_6.jpg)

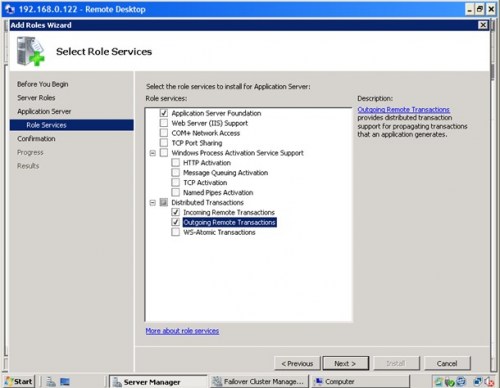
In the **Select Server Roles** dialog box, select the **Application Server** checkbox. This will prompt you to add features required for Application Server role. Click **Next**.

[](http://dbperf.files.wordpress.com/2010/07/1687_71.jpg)

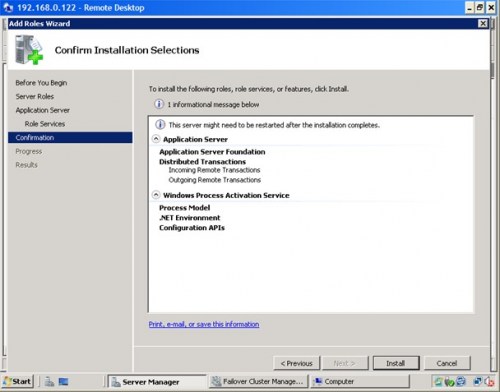
In the **Application Server** dialog box, click **Next**.

[](http://dbperf.files.wordpress.com/2010/07/1687_8.jpg)

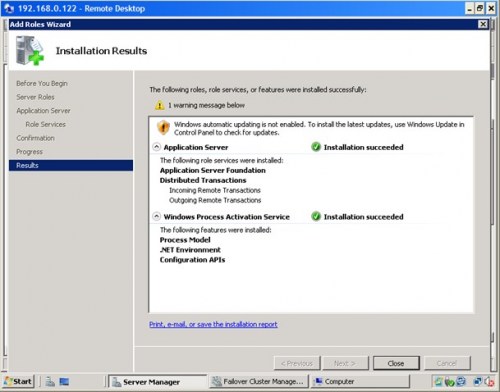
In the **Select Role Services** dialog box, select **Incoming Remote Transactions** and **Outgoing Remote Transactions** checkboxes. These options will be used by MSDTC. Click **Next**

[](http://dbperf.files.wordpress.com/2010/07/1687_9.jpg)

In the **Confirm Installation Selections** dialog box, click **Install**. This will go thru the process of installing the Application Server role

[](http://dbperf.files.wordpress.com/2010/07/1687_10.jpg)

In the **Installation Results** dialog box, click **Close**. This completes the installation of the Application Server role on the first node. You will have to repeat this process for the other server

[](http://dbperf.files.wordpress.com/2010/07/1687_11.jpg)

We have now gone thru the process of creating the cluster at this point. In the next tip in this series, we will go thru the process of installing the Failover Cluster feature, validating the nodes that will become a part of the cluster and creating the cluster itself. And that is just on the Windows side. Once we manage to create a working Windows Server 2008 cluster, that’s the only time we can proceed to install SQL Server 2008.

To continue this series on Installing SQL Server 2008 on a Windows Server 2008 Cluster, we will look at building our Windows Server 2008 cluster in preparation for SQL Server 2008. We have completed the installation of the Application Server role in both of the servers that we will be using as part of our cluster. This tip will walk you through the installation of the Failover Cluster Feature, validating the servers that will be a part of the cluster, and creating the cluster.

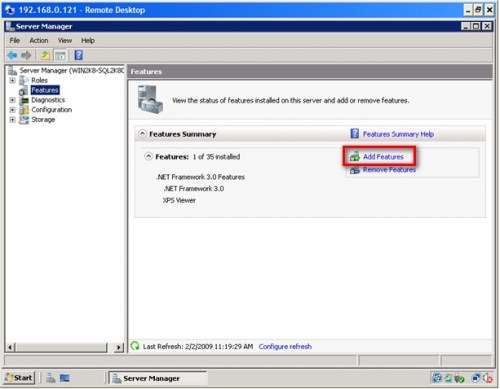
***Adding the Failover Cluster Feature***

Windows Server 2008 calls them features which are simply software programs that can support or augment the functionality of server roles. Since we’ve already installed the Application Server role in our server, let’s define a feature from this perspective: failover clustering simply augments the role as an application server by making it highly available. It is disabled by default, unlike in Windows Server 2003 so we need to add it on both of the servers that will form a part of our cluster.

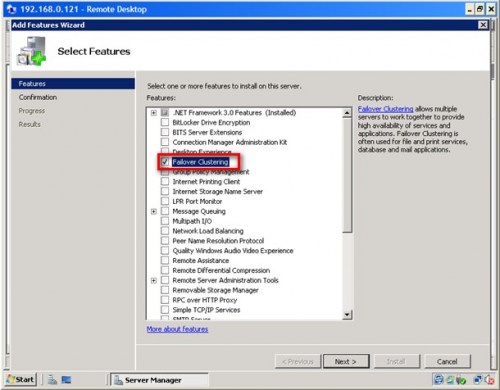
To add the Failover Clustering feature:

Open the **Server Manager** console and select **Features**.

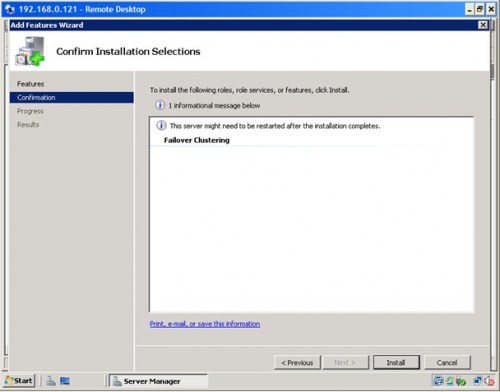
Click the **Add Features** link. This will run the **Add Features** Wizard

[](http://dbperf.files.wordpress.com/2010/07/1698_1.jpg)

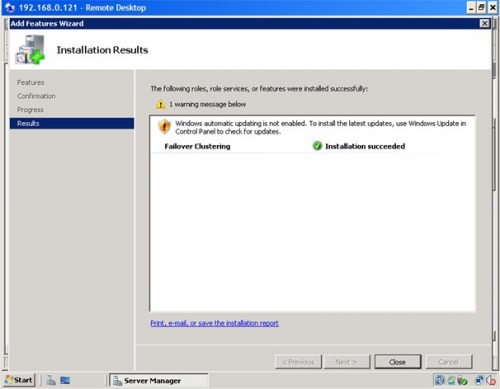
In the **Select Features** dialog box, select the **Failover Clustering** checkbox and click **Next**.

[](http://dbperf.files.wordpress.com/2010/07/1698_2.jpg)

In the **Confirm Installation Selections** dialog box, click **Install** to confirm the selection and proceed to do the installation of the Failover Clustering feature.

[](http://dbperf.files.wordpress.com/2010/07/1698_3.jpg)

In the **Installation Results** dialog box, click **Close**. This completes the installation of the Failover Clustering feature on the first node.

[](http://dbperf.files.wordpress.com/2010/07/1698_4.jpg)

That’s how simple and easy it is to add the Failover Clustering feature in Windows Server 2008. You will have to do this on both nodes to complete the process. Once you have managed to install the Failover Cluster Feature on both nodes, we can proceed to validate our servers if they are ready for clustering.

***Running the Windows Server 2008 Validate Cluster Configuration***

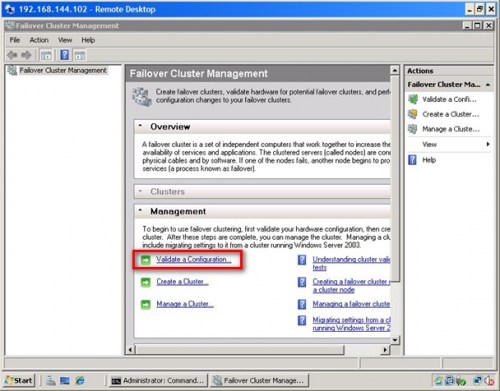
Unlike in previous versions of Windows where Microsoft had some sort of a hardware compatibility list (HCL) from which we had to find and select components tested to be clustering-supported, this wizard is like the “seal” that tells you whether or not the hardware you are using is supported. In fact, Microsoft has partnered with hardware vendors to create the [Failover Cluster Configuration Program](http://www.microsoft.com/windowsserver2008/en/us/clustering-program.aspx) to make the acquisition of hardware for Windows Server 2008 Failover Clustering very easy and simple. Basically, your hardware will be supported for clustering if it meets these two requirements: the server has a “Certified for Windows Server 2008” logo and it passes this wizard.

One word of caution: do not skip any error message that this wizard generates in the final report. Doing so would simply mean that your configuration going forward will be unsupported. You only need to run this wizard on either of the nodes.

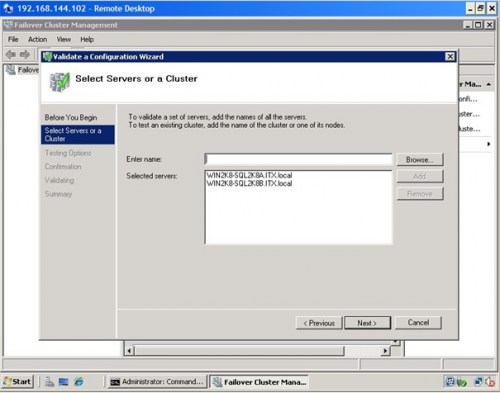
To run the **Validate Cluster Configuration** Wizard:

Open the **Failover Cluster Management** console

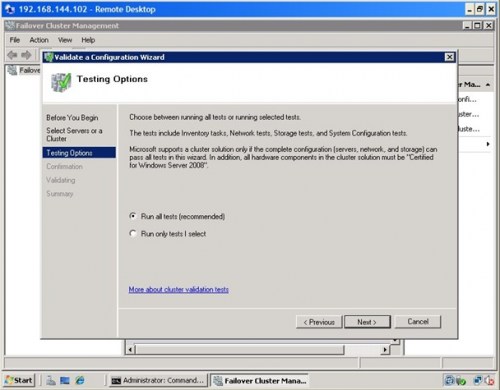
Under the **Management** section, click the **Validate a Configuration** link. This will run the **Validate a Configuration** Wizard

[](http://dbperf.files.wordpress.com/2010/07/1698_5.jpg)

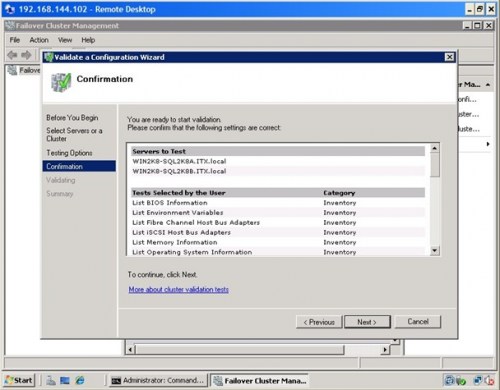
In the **Select Servers or a Cluster** dialog box, enter the hostnames of the nodes that you want to add as members of your cluster and click **Next**.

[](http://dbperf.files.wordpress.com/2010/07/1698_6.jpg)

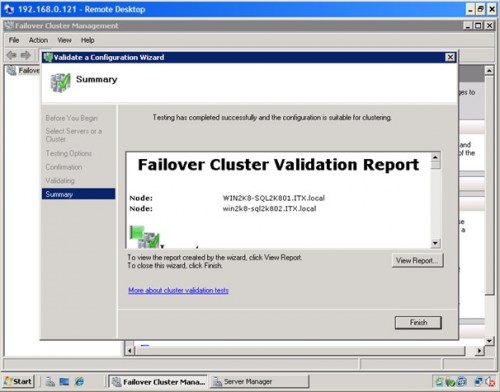
In the **Testing Options** dialog box, click **Next** to run all the necessary tests to validate whether or not the nodes are OK for clustering. If this is the first time to run the wizard, you must run all the tests for validation. For succeding runs, especially when adding hardware like disk subsystems ot network cards on your cluster nodes, you can selectively choose which tests to run as long as you have initially validated your hardware by running all tests.

[](http://dbperf.files.wordpress.com/2010/07/1698_7.jpg)

In the **Confirmation** dialog box, click **Next**. This will run all the necessary validation tests.

[](http://dbperf.files.wordpress.com/2010/07/1698_8.jpg)

In the **Summary** dialog box, verify that all the report returns successful.

[](http://dbperf.files.wordpress.com/2010/07/1698_9.jpg)

If you have reached this part of the process, the wizard will tell you whether or not you can proceed to the next step of creating your cluster. As I’ve mentioned earlier, do not attempt to go any further if this report returned any error messages. I have seen some installations where the shared disk is displaying an error in the validation report prompting me to reconfigure the shared disk. This could mean removing the disk subsystem from both nodes, creating new disks and presenting them on the nodes as mentioned. It would be best to work with your storage engineers or your system administrators when in doubt as different vendors may have different implementations of their disk subsystems.

I’ve also seen issues pertaining to IPv6. This is a fairly common issue which can easily be resolved. The error message in the cluster validation report looks something similar to the one displayed below

Verifying that there are no duplicate IP addresses between any pair of nodes.

Found duplicate IP address fe80::100:7f:fffe%13 on node node1.domain.local adapter

Local Area Connection\* X and node node2.domain.local adapter Local Area Connection\* X.

This [blog post](http://blogs.msdn.com/clustering/archive/2008/07/26/8773796.aspx) outlines the step in resolving this issue. In a few cases, however, I needed to disable the Teredo Tunneling Pseudo-Interface adapter from Device Manager before I got a successful summary report generated by the Validate Cluster Configuration wizard. The bottom line is simply to make sure that the report returns a successful validation before creating the cluster.

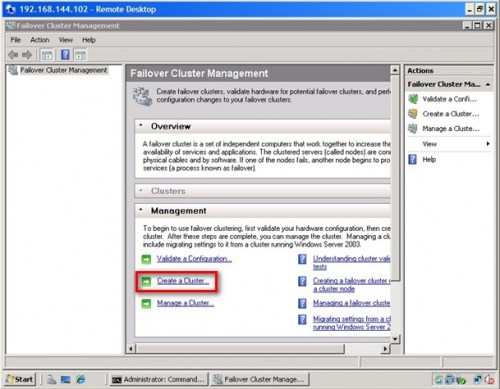
***Creating the Windows Server 2008 Cluster***

You’ve finally reached this step in the entire process. This means you are now ready to create your Windows Server 2008 cluster. It’s as easy as running the Create Cluster Wizard on either of the nodes. Make sure that you have your virtual hostname and IP address ready before proceeding

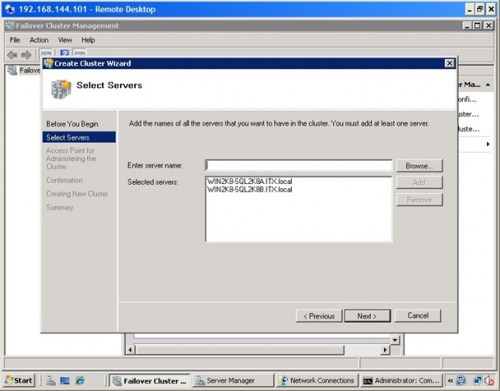
To run the **Create a Cluster** Wizard:

Open the **Failover Cluster Management** console

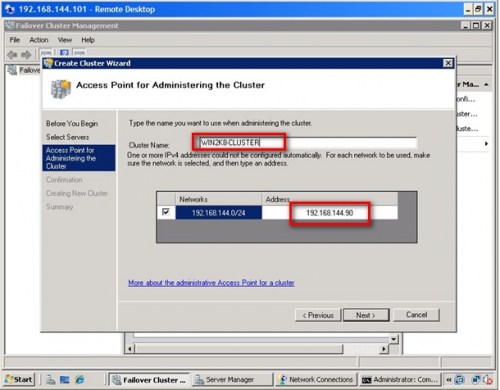
Under the **Management** section, click the **Create a Cluster** link. This will run the **Create Cluster** Wizard

[](http://dbperf.files.wordpress.com/2010/07/1698_10.jpg)

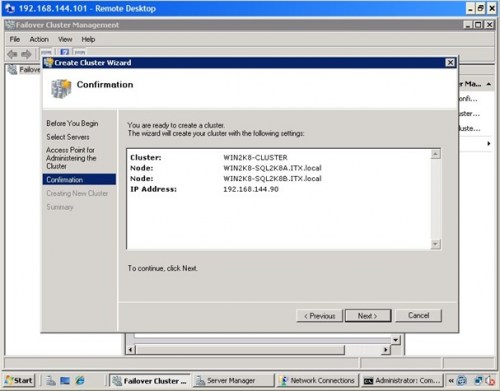
In the **Select Servers** dialog box, enter the hostnames of the nodes that you want to add as members of your cluster and click **Next**.

[](http://dbperf.files.wordpress.com/2010/07/1698_11.jpg)

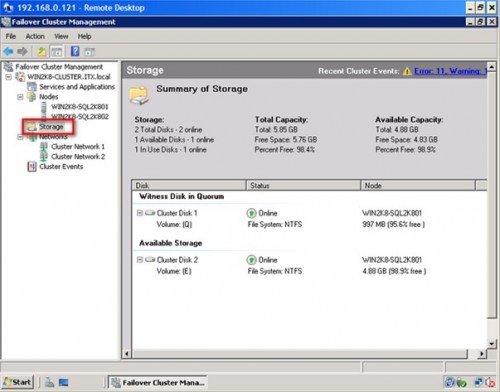
In the **Access Point for Administering the Cluster** dialog box, enter the virtual hostname and IP address that you will use to administer the cluster. Click **Next**

[](http://dbperf.files.wordpress.com/2010/07/1698_12.jpg)

In the **Confirmation** dialog box, click **Next**. This will configure **Failover Clustering** on both nodes of the cluster, add DNS and Active Directory entries for the cluster hostname.

[](http://dbperf.files.wordpress.com/2010/07/1698_13.jpg)

In the **Summary** dialog box, verify that all the report returns successful.

[](http://dbperf.files.wordpress.com/2010/07/1698_14.jpg)

Congratulations! You now have a working Windows Server 2008 cluster. Notice how easy it was to do all of these with fewer steps and less configuration compared to its predecessors. You can now validate whether your cluster is working or not. A simple test would be to do a continuous PING on the virtual hostname or IP address that you have assigned to your cluster. Reboot one of the nodes and see how your PING test responds. At this point, you are now ready to install SQL Server 2008.

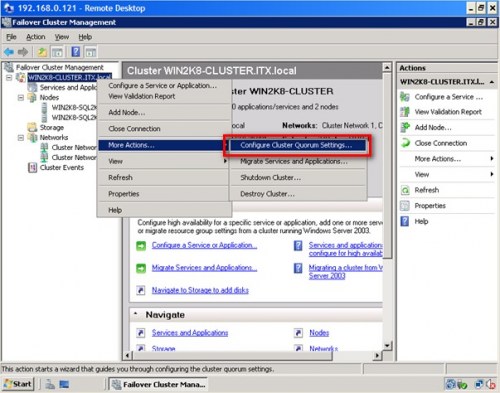
***OPTIONAL: Configuring your cluster quorum***

This section is sometimes necessary especially when Windows Server 2008 decides to take a different disk subsystem as a quorum other than the one you’ve originally intended it to. Notice that in the **Create a Cluster** wizard, there was no option to select the disk subsystem that we can use as a quorum disk (now called the *“witness”* disk). By default, the **Create a Cluster** wizard will use the first available disk as the witness disk. I have seen cases where the originally planned witness disk is sized 1GB while the other shared disks are sized 200 GB. The wizard then selects one of the 200GB-sized disks as a witness disk, requiring you to move the witness disk to the original allocation. To validate, check the **Storage** node under the **Failover Cluster Management** console

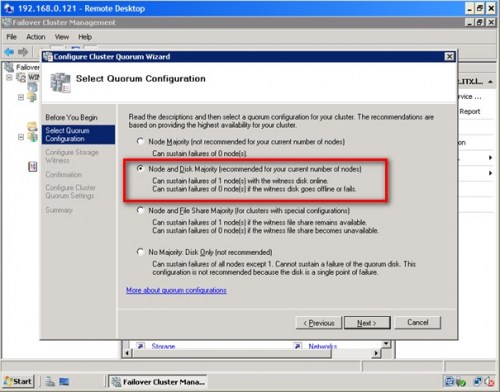
To configure the quorum in a failover cluster:

Open the **Failover Cluster Management** console

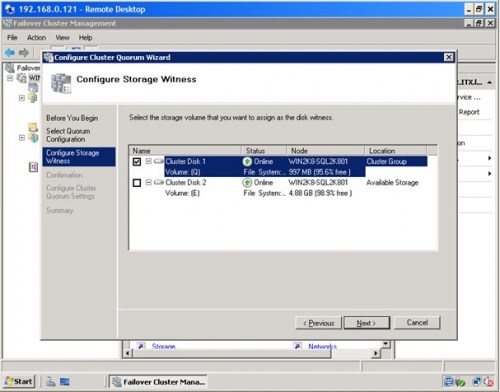
Select the name of the cluster you have just created. Right-click on the cluster, select **More Actions**, and click **Configure Cluster Quorum Settings…** This will open up the **Configure Cluster Quorum** Wizard

[](http://dbperf.files.wordpress.com/2010/07/1698_15.jpg)

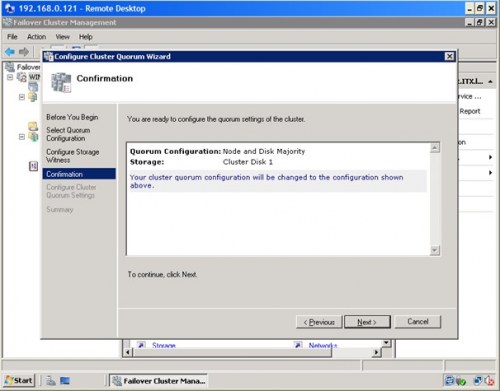
In the **Select Quorum Configuration** dialog box, select the **Node and Disk Majority (recommended for your current number of nodes)** option. The options presented will depend on how you want your cluster configured. The current selection is for a 2-node cluster configuration

[](http://dbperf.files.wordpress.com/2010/07/1698_16.jpg)

In the **Configure Storage Witness** dialog box, validate and select the disk that you want your cluster to use as the quorum/witness disk. Click **Next**

[](http://dbperf.files.wordpress.com/2010/07/1698_17.jpg)

In the **Confirmation** dialog box, verify that the disk configuration for the quorum/witness disk is correct. Click **Next**

[](http://dbperf.files.wordpress.com/2010/07/1698_18.jpg)

In the **Summary** dialog box, verify that all the configurations are successful.

For more details on Configuring the Quorum in a Failover Cluster in Windows Server 2008, check out this [Microsoft TechNet article](http://technet.microsoft.com/en-us/library/cc770620.aspx).

To continue this series on Installing SQL Server 2008 on a Windows Server 2008 Cluster, we will look at installing SQL Server 2008 in a failover cluster. Firstly, we have completed the installation of the Application Server role in both of the servers that we will be using as part of our cluster. Secondly walked you through the installation of the Failover Cluster Feature, validating the servers that will be a part of the cluster, and creating the cluster. In this tip, we will proceed to install SQL Server 2008 in a clustered Windows Server 2008 environment.

***Installing and Configuring MSDTC***

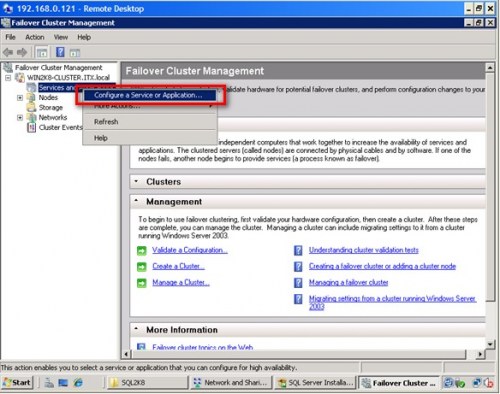
The Microsoft Distributed Transaction Coordinator (MSDTC) is a transaction manager that permits client applications to include several different data sources in one transaction and which then coordinates committing the distributed transaction across all the servers that are enlisted in the transaction. A lot of people ask why we need to install MSDTC prior to installing SQL Server. If you are using distributed transactions or running SQL Server on a cluster, this is definitely a must. SQL Server uses the MSDTC service for distributed queries and two-phase commit transactions, as well as for some replication functionality.

Configuring MS DTC in Windows Server 2003 clusters as defined in this [Microsoft KB](http://support.microsoft.com/kb/301600/) article is not pretty straight-forward. Windows Server 2008 made it simpler by providing a more straightforward process with fewer steps and less configuration.

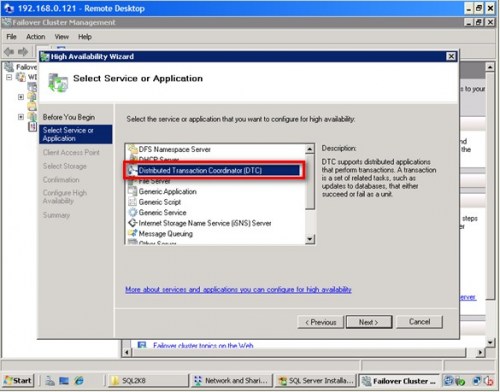
To install and configure MSDTC:

Open the **Failover Cluster Management** console on any of the cluster node.

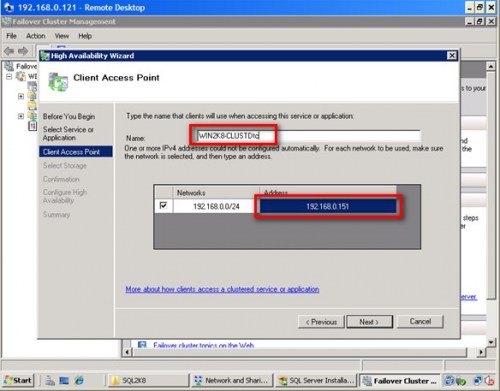
Under the cluster name, right-click on **Server and Applications** and select **Configure a Service or Application**. This will run the **High Availability Wizard**

[](http://dbperf.files.wordpress.com/2010/07/1709_110.jpg)

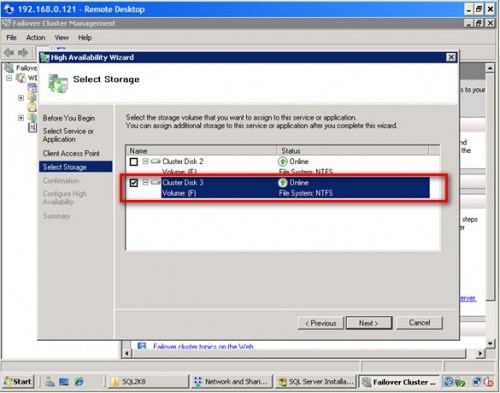
In the **Service or Application** dialog box, select **Distributed Transaction Coordinator (DTC)** and click **Next**.

[](http://dbperf.files.wordpress.com/2010/07/1709_2.jpg)

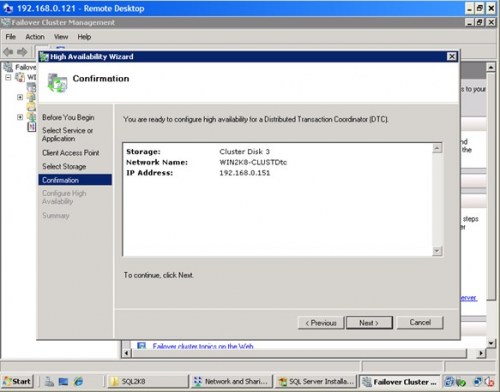
In the **Client Access Point** dialog box, enter the name and IP address of the clustered MSDTC. This should be a different IP addresses and host name from the one that the Windows Server 2008 cluster is already using. Click **Next**.

[](http://dbperf.files.wordpress.com/2010/07/1709_3.jpg)

In the **Select Storage** dialog box, select the disk subsystem that will be used by MSDTC. These disk subsystems have to be defined as available storage in your cluster. In the example below, I have used the disk volume **F:\** and left the disk volume **E:\** for SQL Server later in the installation process. Click **Next**

[](http://dbperf.files.wordpress.com/2010/07/1709_4.jpg)

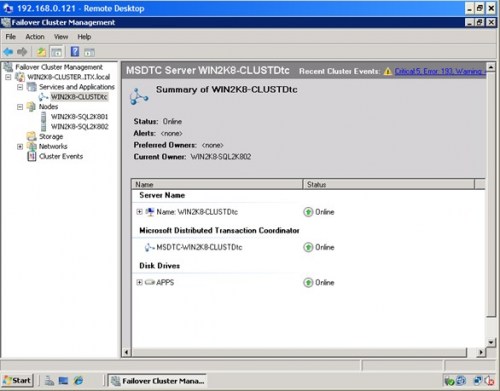
In the **Confirmation** dialog box, validate the configuration you have selected for MSDTC and click **Next**

[](http://dbperf.files.wordpress.com/2010/07/1709_5.jpg)

In the **Summary** dialog box, click **Close**. This completes the installation of MSDTC on the cluster.

[](http://dbperf.files.wordpress.com/2010/07/1709_6.jpg)

You can validate your installation of MSDTC by expanding the **Services and Applications** node and check the cluster name of MSDTC. Make sure that all of the dependency resources are online

[](http://dbperf.files.wordpress.com/2010/07/1709_27.jpg)

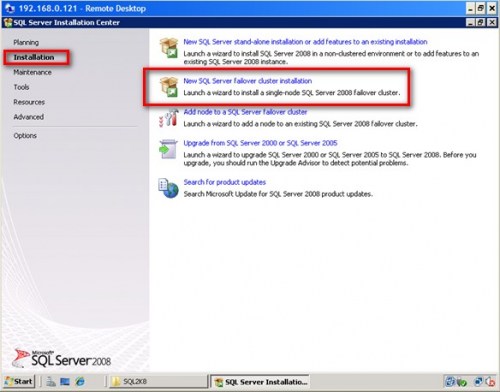
***Installing SQL Server 2008 on a Windows Server 2008 cluster***

You’ve gone this far, don’t stop now. Only after we have managed to prepare everything can we proceed to install SQL Server 2008 on this cluster. Since we’ve already installed [.NET Framework 3.5 with Service Pack 1](http://www.microsoft.com/downloads/details.aspx?familyid=ab99342f-5d1a-413d-8319-81da479ab0d7&displaylang=en) and [Windows Installer 4.5](http://www.microsoft.com/downloadS/details.aspx?familyid=5A58B56F-60B6-4412-95B9-54D056D6F9F4&displaylang=en), we no longer have to worry about them as they both are prerequisites whether you are doing a single server or a cluster installation. There are two options to install SQL Server 2008 on a cluster. The first one is by using the Integrated failover cluster install with Add Node option and the second one is the Advanced/Enterprise installation option. The process outlined below will take into account the first option.

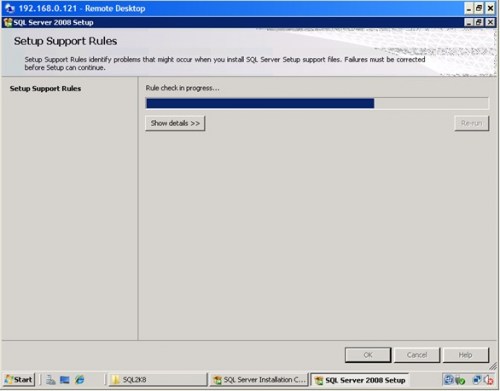
To install SQL Server 2008:

Run **setup.exe** from the installation media to launch **SQL Server Installation Center.** Click on the **Installation** link on the left-hand side

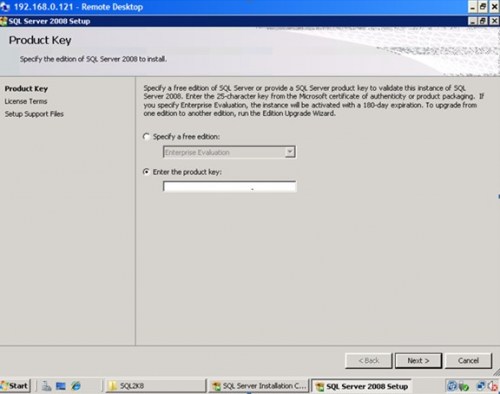
Click the **New SQL Server failover cluster installation** link. This will run the **SQL Server 2008 Setup** wizard

[](http://dbperf.files.wordpress.com/2010/07/1709_7.jpg)

In the **Setup Support Rules** dialog box, validate that the checks return successful results and click **Next**.

[](http://dbperf.files.wordpress.com/2010/07/1709_8.jpg)

In the **Product Key** dialog box, enter the product key that came with your installation media and click **Next**.

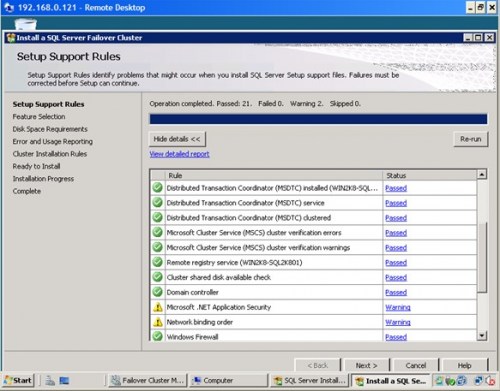
[](http://dbperf.files.wordpress.com/2010/07/1709_9.jpg)

In the **License Terms** dialog box, click the **I accept the license terms** check box and click **Next**. You probably haven’t read one of these, but if you feel inclined go for it.

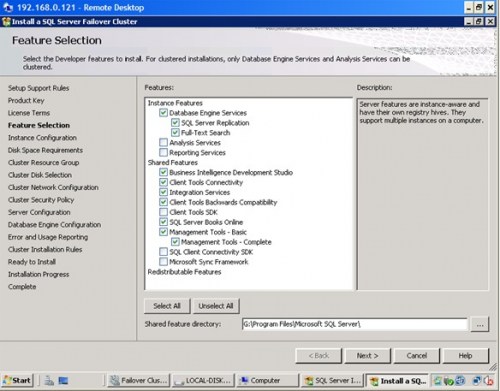
[](http://dbperf.files.wordpress.com/2010/07/1709_10.jpg)

In the **Setup Support Rules** dialog box, click **Install**. Validate that the checks return successful results. If the checks returned a few warnings, make sure you fix them before proceeding with the installation. An example of this is the **Network binding order**. The public network cards should be first on both nodes. Also, you can disable NETBIOS and DNS registration on the network cards to avoid network overhead. Be sure to check your binding order as well. For more details on the network binding order warning, see [Microsoft KB 955963](http://support.microsoft.com/kb/955963).

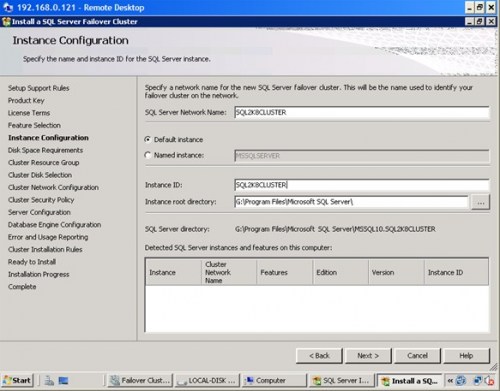
For the Windows Firewall, make sure that you open the appropriate port number on which SQL Server will communicate. You can do this after the installation. Alternatively, you can disable Windows Firewall during the installation and enable it later with the proper configuration. Click **Next** to proceed.

[](http://dbperf.files.wordpress.com/2010/07/1709_11.jpg)

In the **Feature Selection** dialog box, select only the components that you want installed. For the **Shared feature directory**, you can keep the default path if you have sufficient disk space on your C:\ drive or anywhere that is a **local disk** as this will be used by the SQL Server installation process later on. The directory for the clustered database engine will be different. Click **Next**.

[](http://dbperf.files.wordpress.com/2010/07/1709_12.jpg)

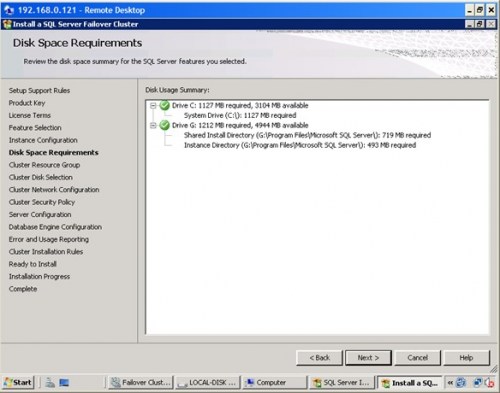
In the **Instance Configuration** dialog box, enter the SQL Server Network Name. This is the name that will be available on the network for the clients. This will vary depending on your selection of whether it is a default or named instance. In this example, default instance is selected.

[](http://dbperf.files.wordpress.com/2010/07/1709_13.jpg)

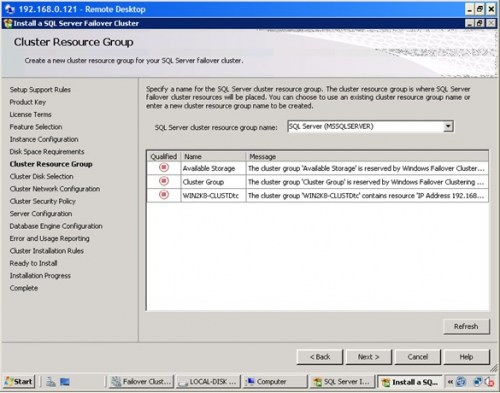
A couple of things need highlighting in this section. By default, the instance name is used as the **Instance ID**. This is used to identify installation directories and registry keys for your instance of SQL Server and is helpful when you want to run multiple instances in a cluster. This is the case 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, you should select the **Instance ID** box and specify a value.

The section on **Detected SQL Server instances and features on this computer** would make sense if there are other SQL Server instances running on your server.

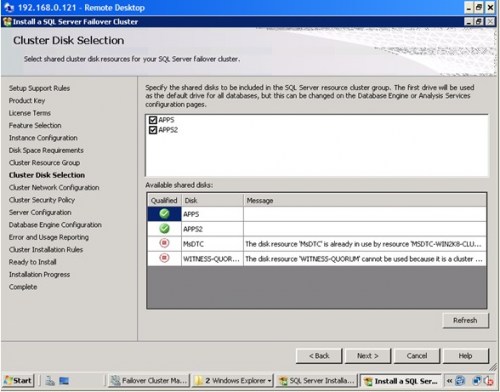
In the **Disk Space Requirements** dialog box, check that you have enough space on your **local disks** to install the SQL Server 2008 binaries and click **Next**.

[](http://dbperf.files.wordpress.com/2010/07/1709_14.jpg)

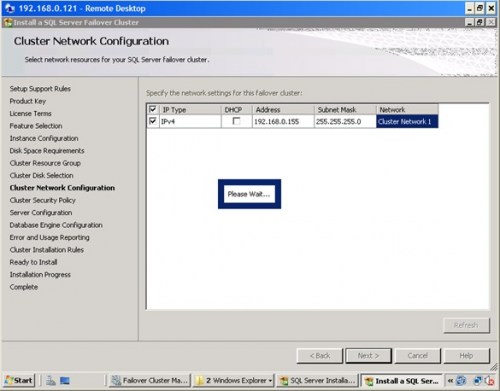
In the **Cluster Resource Group** dialog box, check the resources available on your Windows Server 2008 cluster. This will tell you that a new Resource Group will be created on your cluster for SQL Server. To specify the SQL Server cluster resource group name, you can either use the drop-down box to specify an existing group to use or type the name of a new group to create it. Click **Next**.

[](http://dbperf.files.wordpress.com/2010/07/1709_15.jpg)

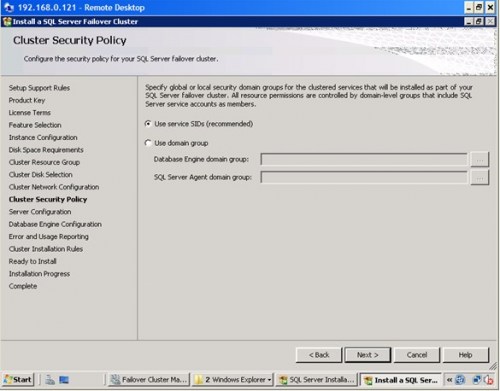
In the **Cluster Disk Selection** dialog box, select the available disk groups that are on the cluster for SQL Server 2008 to use. In this example, two clustered disk groups – **APPS** and **APPS2** – have been selected to be used by SQL Server 2008. I will be using one disk resource for the system databases while the other one for the user databases. Click **Next**.

[](http://dbperf.files.wordpress.com/2010/07/1709_16.jpg)

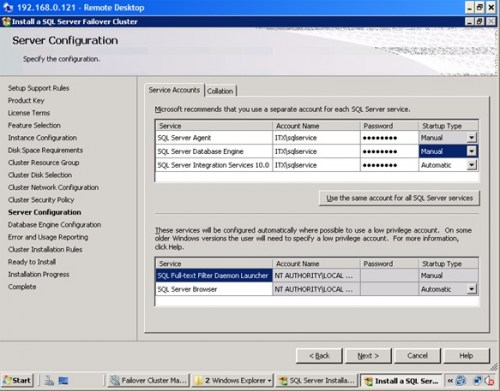
In the **Cluster Network Configuration** dialog box, enter the IP address and subnet mask that your SQL Server 2008 cluster will use. Deselect the checkbox under the **DHCP** column as you will be using static IP addresses. If you have not disabled your IPv6 adapters and protocols, it would be better to uncheck the row for **IPv6**

[](http://dbperf.files.wordpress.com/2010/07/1709_17.jpg)

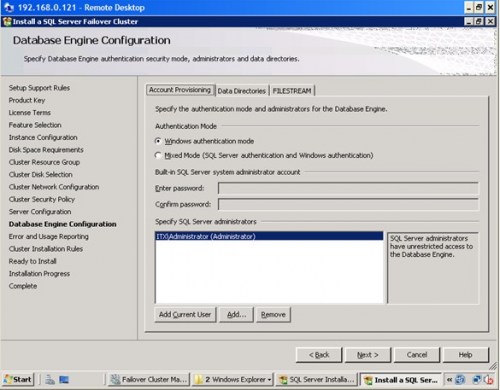
In the **Cluster Security Policy** dialog box, accept the default value of **Use service SIDs (recommended)**. In Windows Server 2003, we specify domain groups for all SQL Server services but in Windows Server 2008, this is the recommended option. For more information on using service SIDs for SQL Server 2008, check out this [MSDN article](http://msdn.microsoft.com/en-us/library/ms143504.aspx#Service_SID)

[](http://dbperf.files.wordpress.com/2010/07/1709_18.jpg)

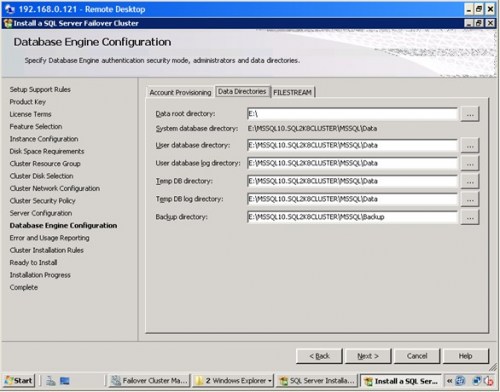
In the **Server Configuration** dialog box, enter the credentials that you will use for your SQL Server service accounts in the **Service Accounts** tab. In the **Collation** tab, select the appropriate collation to be used by SQL Server. Note that the startup type is set to manual for all cluster-aware services and cannot be changed during the installation process. Click **Next**.

[](http://dbperf.files.wordpress.com/2010/07/1709_19.jpg)

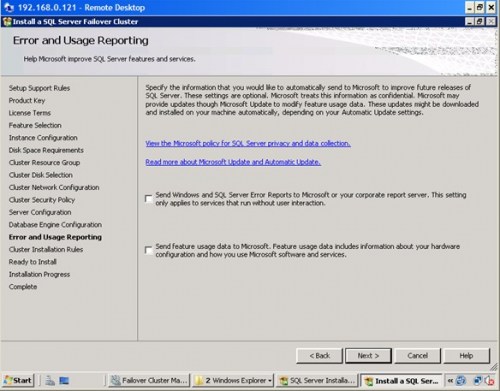
In the **Database Engine Configuration** dialog box, select the appropriate **Authentication Mode.** If you want to add the currently logged on user to be a part of the SQL Server administrators group, click the **Add Current User** button.

[](http://dbperf.files.wordpress.com/2010/07/1709_20.jpg)

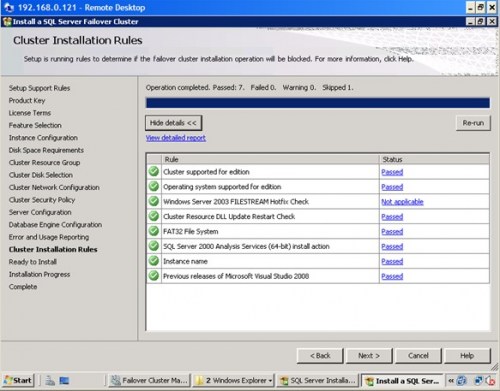
On the **Data Directories** tab, enter the path where your system and user database files will be created. This will default to the first shared disk in the cluster so in case you want to change it to the other shared disks to be used by SQL Server 2008, modify accordingly. If you intend to use the new FILESTREAM feature, click the **FILESTREAM** tab and set the appropriate configurations. Click **Next**

**[](http://dbperf.files.wordpress.com/2010/07/1709_211.jpg)**

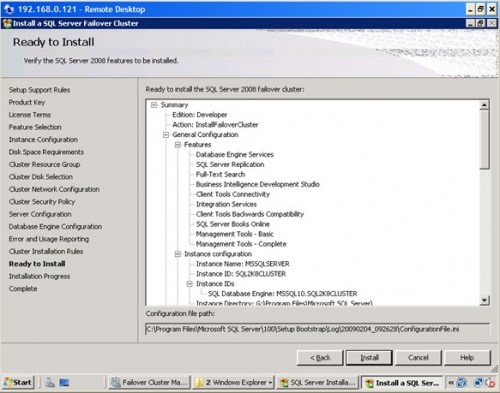
In the **Error and Usage Reporting** dialog box, click **Next**.

[](http://dbperf.files.wordpress.com/2010/07/1709_22.jpg)

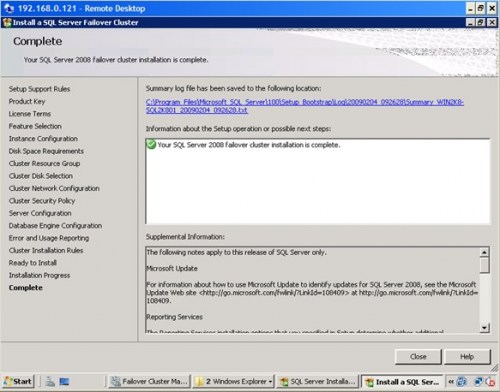
In the **Cluster Installation Rules** dialog box, verify that all checks are successful and click **Next**.

[](http://dbperf.files.wordpress.com/2010/07/1709_23.jpg)

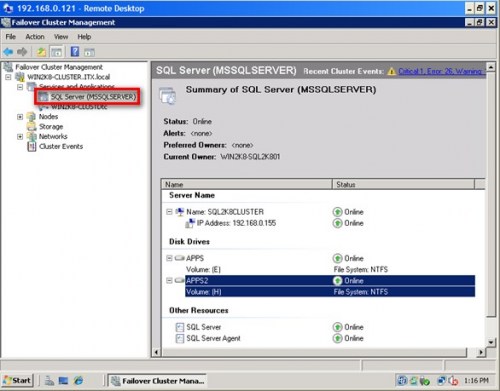
In the **Ready to Install** dialog box, verify that all configurations are correct. Click **Next**.

[](http://dbperf.files.wordpress.com/2010/07/1709_24.jpg)

In the **Complete** dialog box, click **Close**. This concludes the installation of a SQL Server 2008 Failover Cluster

[](http://dbperf.files.wordpress.com/2010/07/1709_25.jpg)

At the completion of a successful installation and configuration of the node, you now have a fully functional failover cluster instance. To validate, open the **Failover Cluster Management** console, and click on **SQL Server (MSSQLSERVER)** under **Services and Applications**. Make sure that all dependencies are online

[](http://dbperf.files.wordpress.com/2010/07/1709_26.jpg)

Although we do have a fully functioning SQL Server 2008 failover cluster, it does not have high-availability at this point in time because there is only one node in the failover cluster. We still have to add the second node to the SQL Server 2008 cluster. In the last part of this series, we will add the second node in the failover cluster and install the latest cumulative update

To continue this series on Installing SQL Server 2008 on a Windows Server 2008 Cluster, we will look at adding a node in a SQL Server 2008 failover cluster. In this third step, we will proceed to add a node in a SQL Server 2008 failover cluster and apply the latest cumulative updates.

***Adding a node on a SQL Server 2008 Failover Cluster***

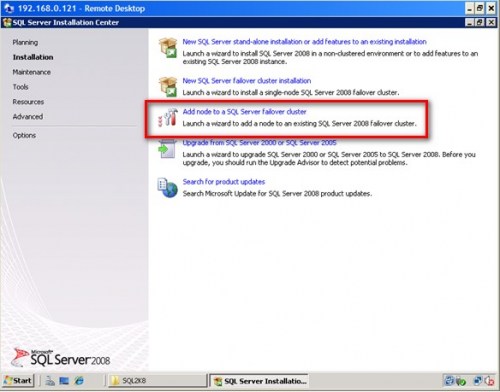
Now that you have a working failover cluster, we will make it highly available by adding nodes. The number of nodes you can add in a failover cluster depends on the editions of SQL Server that you will use. A Standard Edition of SQL Server 2008 can support up to two (2) nodes in a failover cluster while the Enterprise Edition supports up to sixteen (16) nodes, which is practically the limit for the Enterprise Edition for Windows Server 2008. As most of the steps in this process are similar to the one when you were installing the failover cluster, I’ve skipped most of the screenshots.

To add a node on a SQL Server 2008 failover cluster:

Run **setup.exe** from the installation media to launch **SQL Server Installation Center**

Click on the **Installation** link on the left-hand side. Click the **Add node to a SQL Server failover cluster** link. This will run the **SQL Server 2008 Setup** wizard.

There are a couple of glitches when you get to this point. One of them is a popup error with an error message “*failed to retrieve data for this request*” while in this step. I’ve seen a [Microsoft Connect](https://connect.microsoft.com/SQLServer/feedback/ViewFeedback.aspx?FeedbackID=337981) item on this but refers to CTP6 so I was thinking it has already been resolved. After a few searches and questions asked, [SQL Server MVP Geoff Hiten](https://mvp.support.microsoft.com/profile=C482792D-D259-43F5-8128-194C643EF5F2) advised that prior to adding another node in the cluster, any cumulative update should be pre-applied to the node before the main installation as the cluster install of the RTM version has some bugs. This creates a patched install script for the RTM installer to use. The fix started with cumulative update 1 so, technically, you can apply any cumulative update. Sounds weird, but it works. You still have to apply the patch after the installation.

[](http://dbperf.files.wordpress.com/2010/07/1721_1.jpg)

In the **Setup Support Rules** dialog box, validate that the checks return successful results and click **OK**.

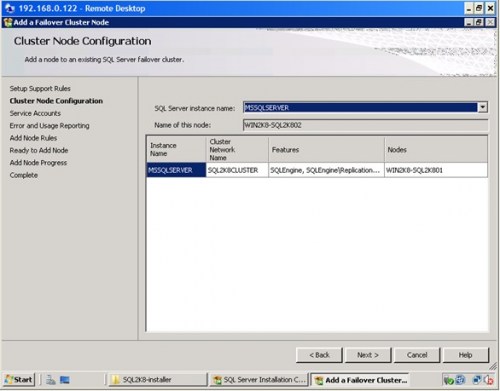
In the **Product Key** dialog box, enter the product key that came with your installation media and click **Next**.

Again, a few glitches on this step. This might seem unusual as you are only being asked about the Product Key. There is also a [Microsoft Connect](https://connect.microsoft.com/SQLServer/feedback/ViewFeedback.aspx?FeedbackID=363036) item for this which basically asks you to run the **setup.exe** in command prompt. There is a popup error with an error message “*The current SKU is invalid*” while in this step. This usually happens when you use a media with a supplied product key, like the one that comes with an MSDN subscription. What worked for me was to copy the installation media on a local disk, locate the file **DefaultSetup.ini** file from the installation files and delete it or move it to different location. If you opt to delete the file, make sure you note down the product key written on this file as you will need to manually key this in during the installation process. This [forum post](http://forums.techarena.in/server-cluster/1032365.htm) will give you quite a few options to solve this issue

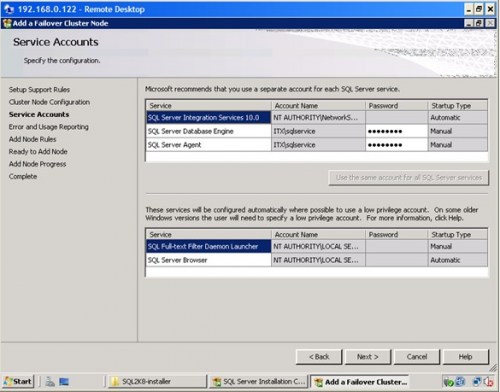
In the **License Terms** dialog box, click the **I accept the license terms** check box and click **Next**.

In the **Setup Support Rules** dialog box, click **Install**. Validate that the checks return successful results. Again, make sure to fix any errors returned by this check before proceeding with the installation.

In the **Cluster Node Configuration** dialog box, validate that the information for the existing SQL Server 2008 cluster is correct.

[](http://dbperf.files.wordpress.com/2010/07/1721_2.jpg)

In the **Service Accounts** dialog box, verify that the information is the same as what you have used to configure the first node.

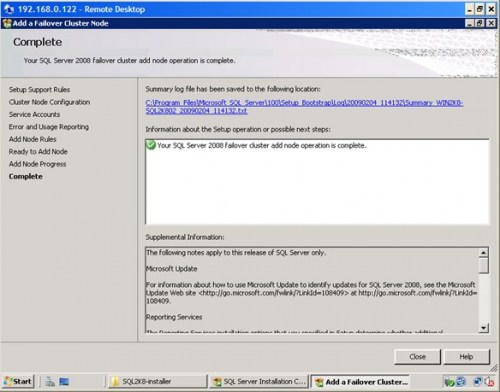
[](http://dbperf.files.wordpress.com/2010/07/1721_3.jpg)

In the **Error and Usage Reporting** dialog box, click **Next**

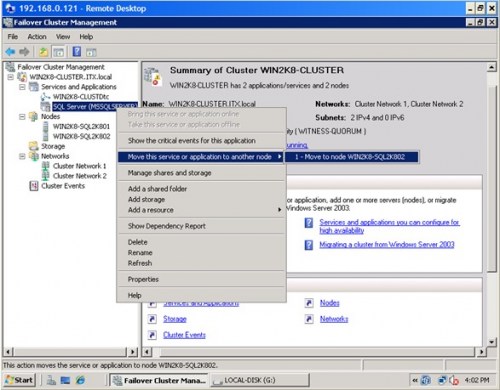
In the **Add Node Rules** dialog box, verify that all checks are successful and click **Next**

In the **Ready to Add Node** dialog box, verify that all configurations are correct and click **Install**

In the **Complete** dialog box, click **Close**. This concludes adding a node to a SQL Server 2008 Failover Cluster

[](http://dbperf.files.wordpress.com/2010/07/1721_4.jpg)

You can validate your cluster installation by expanding the **Services and Applications** node and check the cluster name of your SQL Server instance. You can now see an option to move the service to another node, in this case, the node you’ve just added in your failover cluster

[](http://dbperf.files.wordpress.com/2010/07/1721_5.jpg)

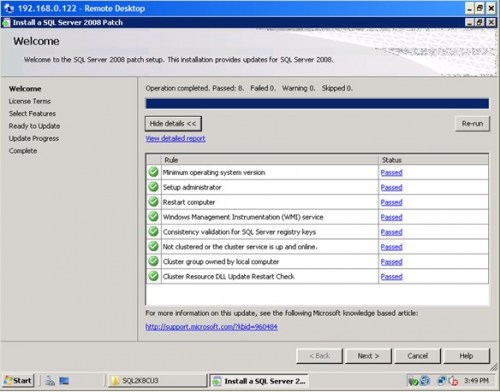
***Applying patches on a SQL Server 2008 cluster***

Part of the tasks of a DBA is to apply patches on the database engine and a SQL Server 2008 failover cluster is no exception. In fact, it is not as straight-forward as applying patches and service packs on a stand-alone server. It is important to note that when applying patches or service packs to a SQL Server failover cluster, you should apply them first on the passive node. After completing the installation on the passive node, failover the SQL Server 2008 cluster resource to this node making it the active node. Once the SQL Server service and all other dependencies are up, you can, then, apply the patches on the new passive node. The latest available patch for SQL Server 2008 is cumulative update 4 and is available for request from Microsoft. For more information, check out this [Microsoft KB article](http://support.microsoft.com/kb/963036/). You will have to request for the patch from Microsoft as it is not available from the Microsoft Download Center. The screenshots below show cumulative update 3 (version **10.0.1600.22**) but the process is basically the same. Also, note that even though you may have already applied the cumulative update due to the bug mentioned above for adding a node in a failover cluster, you still have to apply the patch on both nodes

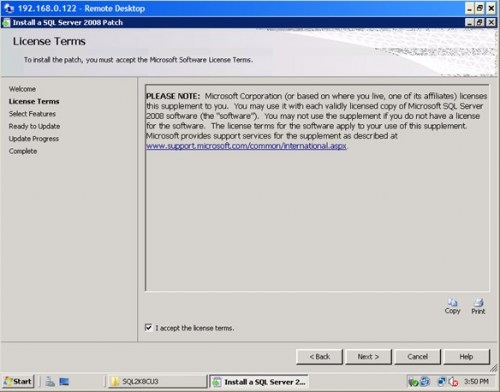
To apply patches on a SQL Server 2008 failover cluster node:

Run **SQLServer2008-KB960484-x86.exe** (this would depend on the cumulative update that you want to apply) from the hotfix package you have requested from Microsoft

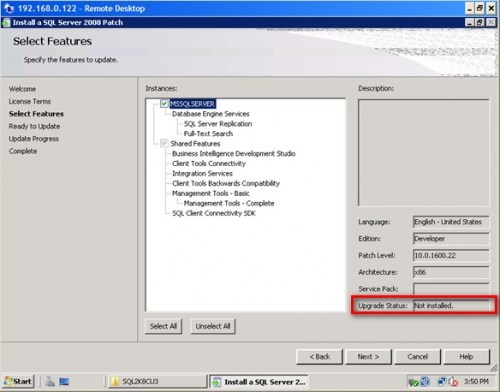
In the **Welcome** dialog box, validate that the checks return successful results.

[](http://dbperf.files.wordpress.com/2010/07/1721_6.jpg)

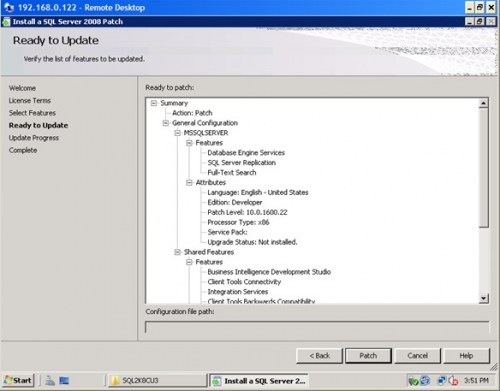
In the **License Terms** dialog box, click the **I accept the license terms** check box and click **Next**

[](http://dbperf.files.wordpress.com/2010/07/1721_7.jpg)

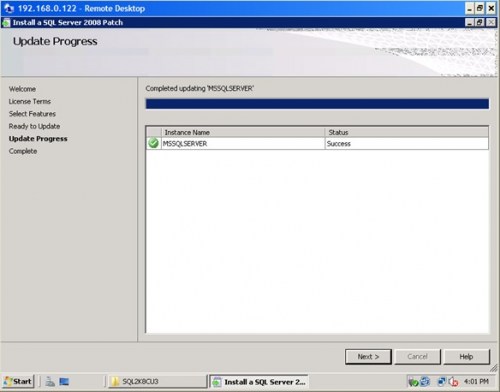
In the **Select Features** dialog box, validate the SQL Server 2008 components by clicking on the check box. The **Upgrade Status** field will tell you whether or not the patch has already been applied. Click **Next**

[](http://dbperf.files.wordpress.com/2010/07/1721_8.jpg)

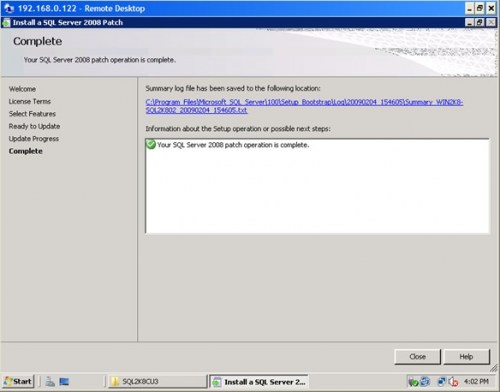
In the **Ready to Update** dialog box, verify that all configurations are correct and click **Patch**

[](http://dbperf.files.wordpress.com/2010/07/1721_9.jpg)

In the **Update Progress** dialog box, validate that the installation was successful.

[](http://dbperf.files.wordpress.com/2010/07/1721_10.jpg)

In the **Complete** dialog box, click Close. This concludes patching the passive node of a SQL Server 2008 Failover Cluster

[](http://dbperf.files.wordpress.com/2010/07/1721_11.jpg)

After successfully installing the patch on the passive node, move the SQL Server 2008 cluster resource to this node so it will become the new active node. Make sure that all the SQL Server 2008 cluster dependencies are online prior to applying the patch on the other node. Repeat the process outlined above to the new passive node. A more comprehensive approach for applying a SQL Server 2008 patch to a failover cluster instance is defined in this [Microsoft KB article](http://support.microsoft.com/default.aspx/kb/958734)

Congratulations! You now have a working two-node SQL Server 2008 failover cluster running on Windows Server 2008.