

# Applying Updates to a Clustered Instance of SQL Server 2008 or SQL Server 2008 R2

---

Written by  
Allan Hirt  
Cluster MVP  
Megahirtz LLC

# Contents

---

- Abstract..... 2
- Introduction..... 3
- The New Update Process..... 4
- Shared versus Dedicated Components..... 6
- Scenario One – No Dedicated Failover Node ..... 7
- Detailed Steps for Scenario One ..... 24
- Conclusion for Scenario One..... 26
- Scenario Two – Dedicated Failover Node ..... 27
- Detailed Steps for Scenario Two ..... 30
- Conclusion for Scenario Two..... 32
- About the Author..... 33

# Abstract

---

Applying updates to a clustered SQL Server configuration where there are multiple instances on the same failover cluster can be complicated. This whitepaper uses two sample scenarios to show you how to approach and execute the installation of updates to clustered instances.

# Introduction

---

The components of a computer, including the software that runs on it, are not unlike a car: At some point they will need some kind of maintenance. Software maintenance usually takes the form of an update or patch, which remains supported by the vendor. It can also involve fixing a specific issue that was found by the vendor or encountered by those using the application.

Staying with a supported product version and maintaining stability are both critical to achieving high availability. Still, you'll occasionally need to incur short periods of downtime in order to have longer overall uptime. For SQL Server, this means installing periodic service packs (SPs), community updates (CU), or even one-off hotfixes, as required. For more information on developing a lifecycle for product versions, see the [white paper "Testing and Developing Supportability Roadmaps for Independent Software Vendor Applications."](#) Ignore the fact that it references ISVs in the title; the information in there applies to anyone running SQL Server.

# The New Update Process

---

The way updates are applied to a clustered SQL Server instance changed with SQL Server 2008. The patching process closely follows how the new setup process works for a new SQL Server 2008 or SQL Server 2008 R2 clustered instance: It is essentially a node-by-node affair with a few minor differences. The redesign prevents the occasional failure that could happen in previous versions, where one node would not receive an updated file, causing problems later.

Here is the new upgrade process at a high level:

1. Before you do any installations of the SQL Server update, you need to reconfigure an instance's network name resource to remove the node(s) that will be patched. Reconfiguring the network name prevents the resource group from failing over to another node that has not yet been upgraded, thereby keeping everything consistent. The removed nodes are not currently running that SQL Server instance. This allows you to remain up and running until later in the process.

Microsoft recommends removing at least half of the nodes that are possible owners of the network name resource for the instance that will be patched. Here's the way to approach it: Think of the nodes for each instance as a set or a specific group of servers. They should be about equal (or rounded up) in the case of an odd number of nodes.

The number of nodes removed may be slightly more than half, in the case of an odd number. Think about it like a set: Each instance has N nodes, which are part of its definition. You need to remove enough members of the set for it to meet Microsoft requirements.

One caveat: In some cases (such as with three nodes), you'll end up removing two — that way there's no place to fail over to while the other nodes are being updated.

2. Patch the nodes that were removed as part of the set.
3. Add the newly patched nodes back into the set.
4. Move the instance, which is still currently at the old patch level, to one of the newly patched nodes. This will trigger the upgrade of the instance itself. The upgrade process will be shown in the SQL Server log. (A sample can be found in the downloadable files for this white paper.)
5. Verify that the instance upgrade was successful.
6. Remove the node(s) not yet patched from the set.
7. Patch the newly removed nodes.
8. Add the newly patched node(s) back into the set.
9. Test failover of the upgraded instance to all nodes to ensure that in the event of a problem, all nodes can own it.
10. Test applications against the upgraded instance to ensure they still function properly.
11. Allow applications and end users to use the newly upgraded instance.

The challenge comes when you have more than two nodes and a single instance. Then the process for installing an update to those clustered instances becomes a bit more complicated, as you'll need to worry about things like reboots and resource contention. The main difference from the new install (and the add node operation as well) is that a patch installation can be applied to multiple instances at once. This gives you some options, depending on which path you take based on the number of instances and nodes you have.

During the design of the cluster, the failover scenarios should always be taken into account. There should be enough capacity on the nodes to run more than one (or possibly all) of the instances at any given time, not just one. If this has not been taken into account, then upgrading a multi-node SQL Server failover cluster could pose some challenges for your environment. Each scenario presented here assumes that one node can run more than a single instance.

You are trying to prevent three scenarios from happening during an update of clustered instances:

1. Do not allow an instance that has not been upgraded to be able to run a node that is in the process of being patched. This is common sense: If files are being updated, even if it means the instance has nowhere to go, you do not want it running somewhere that is in flux.
2. After the instance upgrade occurs, there may still be nodes that need to be upgraded. Do not allow an uplevel instance to be owned by a node that has downlevel binaries. While this may work and you may see no apparent issues, it is definitely not recommended.
3. Do not lose quorum. A failover cluster has a quorum when the majority of its owners are up and running. Even if a node needs to be rebooted as part of the patching, if you reboot too many nodes at a given time you could lose the quorum and shut down the entire Windows failover cluster. This would create an unintended outage for anything running in the cluster, and this is why the steps need to be executed very carefully in this example. For more on quorum in a cluster and the different quorum modes, see [this article](#) in TechNet.

This whitepaper will show two examples of how to achieve a successful patching of a multi-node, multi-instance cluster. For the most basic scenario of one instance on two nodes, see my blog post. The examples have failover clusters with different configurations. Each scenario accounts for everything mentioned above to maximize uptime where possible and minimize the amount of work that needs to be done. Both example scenarios were tested. The files will be available for download at <http://www.sqlha.com/resources>.

# Shared versus Dedicated Components

When you install SQL Server, you get a mixture of components shared by all instances that may be installed as well as some dedicated ones, which are used only by a particular instance. So how can you tell which is a shared component, and which is dedicated to an instance? Under the top-level folder that contains the SQL Server binaries on each node (usually *C:\Program Files\Microsoft SQL Server*), you will see another series of folders. In Figure 1, you will see 80, 90, and 100 folders. Those contain the shared files that are used by all of the instances. The other folders contain the binaries for a specific instance of SQL Server. For example *MSSQL10.GRAND\_ILLUSION* is the folder containing the binaries for instance *GRANDILLUSION*.

**TIP:** The name of the folder past *MSSQL10* is set during installation using the value for Instance ID. It cannot be altered post-setup. For a clustered instance, you should set this value to something recognizable because it will be the same on each node. You'll want to be able to discern between the instances.

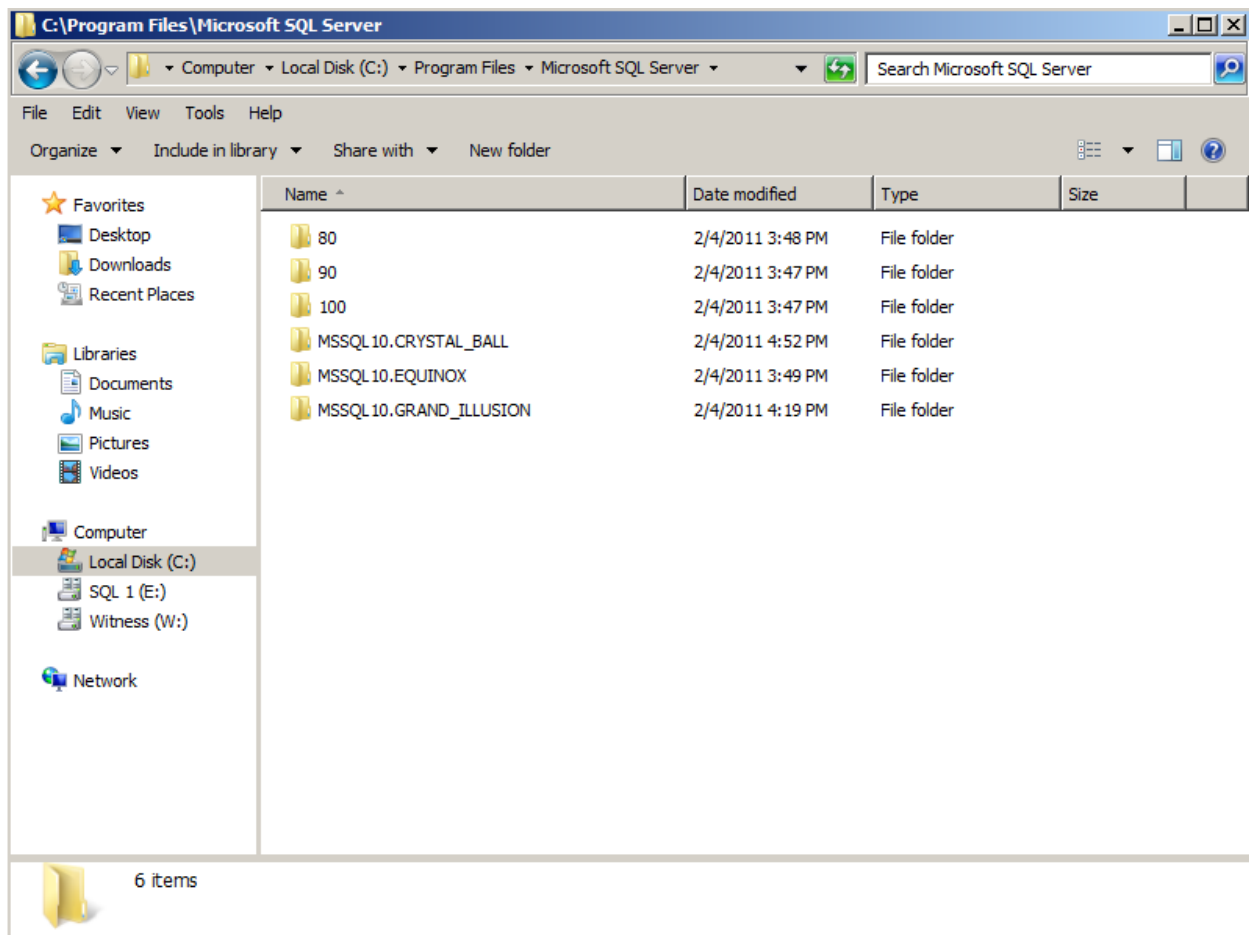


Figure 1. Folder structure for SQL Server binaries

# Scenario One – No Dedicated Failover Node

This first scenario will walk through applying SQL Server 2008 SP2 to a failover cluster that has three instances (all at SQL Server 2008 SP1) spread out over three nodes. This is a fairly common scenario in many environments since some think that having a node that is just for failover is considered “wasting” resources. It is not uncommon to see at least one instance per node. The initial state of the cluster is shown in Table 1.

The approach in this scenario was to minimize the time it takes to do the process. Many reading this paper have short maintenance windows. This would be considered a more aggressive approach. A more conservative approach would be to patch instance-by-instance and potentially spread the work of patching the different instances over multiple outage windows. What is shown in this scenario is only one possibility and path you could take.

Here’s how to read the tables in this document:

- As changes occur to the cluster configuration they will be noted in italicized red in the tables.
- The Owner column means that node is currently the one with the resources for that SQL Server instance.
- The Possible Owner shows the status of the network name resource and which nodes are currently configured as possible owners.
- The Version column shows the status of the upgrade for a particular instance on each node.

	Node A			Node B			Node C		
	Owner	Possible Owner	Version	Owner	Possible Owner	Version	Owner	Possible Owner	Version
<b>Instance 1</b>	Yes	Yes	SP1	No	Yes	SP1	No	Yes	SP1
<b>Instance 2</b>	No	Yes	SP1	Yes	Yes	SP1	No	Yes	SP1
<b>Instance 3</b>	No	Yes	SP1	No	Yes	SP1	Yes	Yes	SP1

**Table 1. Initial cluster state**

In this example, Instance 1 will be the first one to be upgraded to SP2. Since there are only three nodes, following Microsoft’s guidelines, Node B and Node C must be removed as possible owners of the network name resource associated with Instance 1. It is important to note that removing possible owners is a dynamic setting for a cluster resource and does not require any downtime. The cluster configuration now looks like the one shown in Table 2.

Removing those nodes means that Instance 1 will have nowhere to move should a problem occur while Nodes B and C are being patched. This should be considered an acceptable risk because there is a good probability nothing will happen to the running instance and should a problem occur during the installation of the update, the failure is completely isolated. Until Node B and Node C are patched for Instance 1, Instance 1 will remain up to service requests.

**TIP:** Because this process is going to be repeated many times, you will need to keep track of which nodes are in or out as possible owners, so script the removal and addition of nodes as possible owners to the network name resource. These scripts can be performed using the command line cluster.exe (Windows Server 2003, Windows Server 2008, Windows Server 2008 R2; it has been deprecated as of Windows Server 2008 R2) or PowerShell (Windows Server 2008 R2 only). The sample scripts all use PowerShell. For more information on using both cluster.exe and PowerShell, see my blog post “[Scripting the Addition and Removal of Cluster Nodes as Possible Owners of a Resource.](#)”



	Node A			Node B			Node C		
	Owner	Possible Owner	Version	Owner	Possible Owner	Version	Owner	Possible Owner	Version
<b>Instance 1</b>	Yes	Yes	SP1	No	No	SP1	No	No	SP1
<b>Instance 2</b>	No	Yes	SP1	Yes	Yes	SP1	No	Yes	SP1
<b>Instance 3</b>	No	Yes	SP1	No	Yes	SP1	Yes	Yes	SP1

**Table 2. State of the cluster prior to upgrading Instance 1**

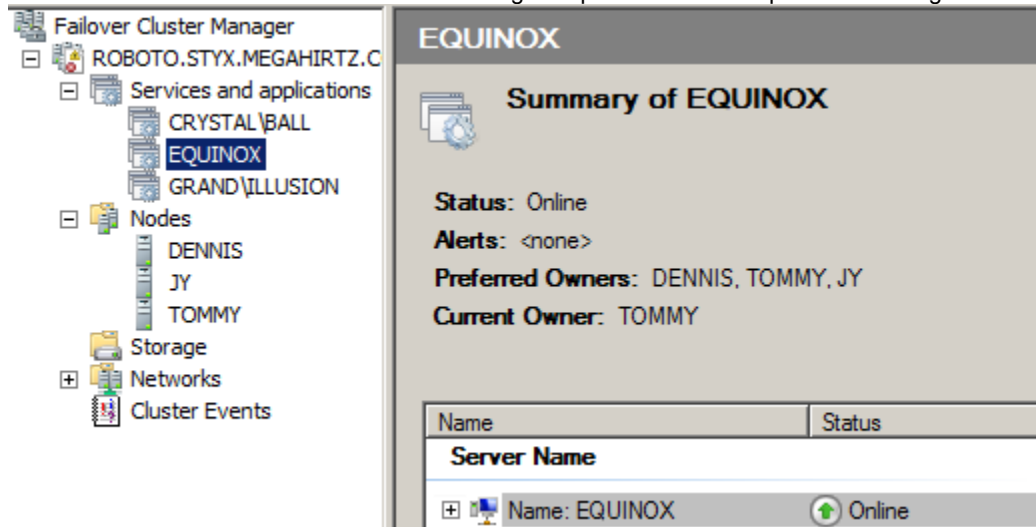
You can check the possible owners either via the script output or in Failover Cluster Manager. Figure 2 shows the output of the corresponding PowerShell commands, which show the before and after status of the network name resource. The first line shows that the network name resource for an instance named EQUINOX has all three nodes as possible owners. After the command to modify the configuration, the output shows only one node (DENNIS) can own the resource.

```
ClusterObject                                OwnerNodes
-----
SQL Network Name <EQUINOX>                  {dennis, jy, tommy}
SQL Network Name <EQUINOX>                  {dennis}
```

**Figure 2. Demonstrating the possible owner change in PowerShell**

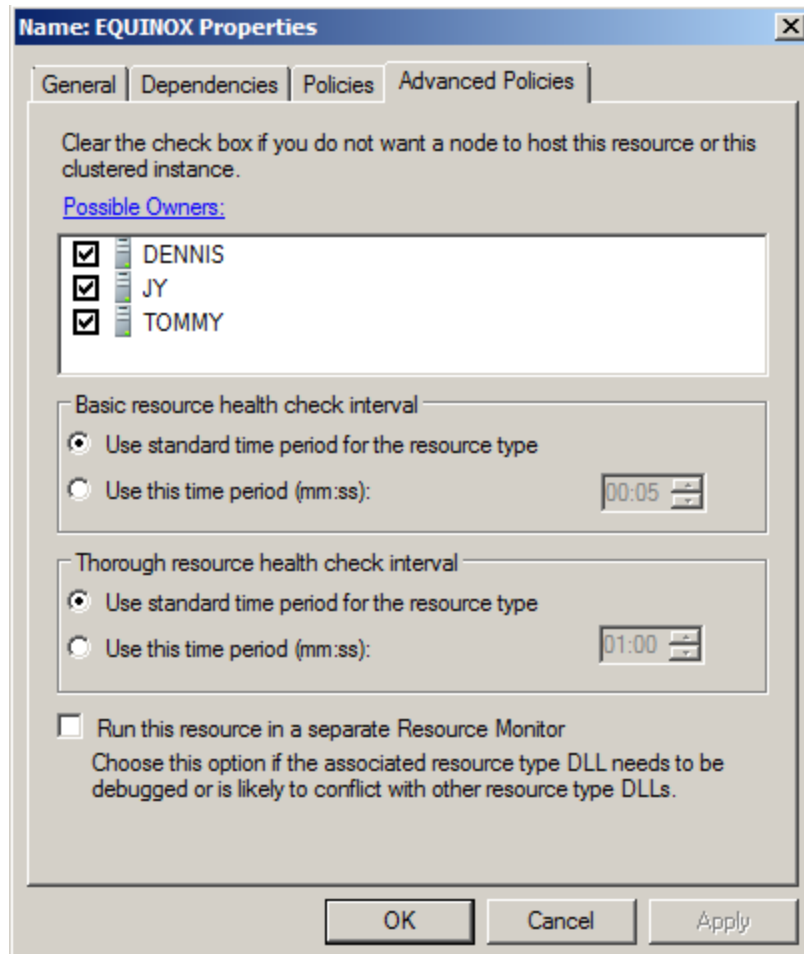
To check the possible owners in Failover Cluster Manager, do the following:

1. Navigate to the resource group/application that contains the SQL Server instance.
2. Select the Name: <Network Name You Gave It During Setup>. See the example shown in Figure 3.



**Figure 3. Sample Network Name resource (Name: EQUINOX)**

3. Right click on it, and select Properties.
4. Select the Advanced Policies tab. You will see the list of possible owners, as shown in Figure 4. Any node that has a check next to it can be a possible owner.



**Figure 4. Advanced Policies tab for the network name resource**

Before SP2 can be installed against Instance 1 on Node B, you must fail Instance 2 to another node as shown in Table 3. Since Instance 2 is already starting to be interrupted, the best thing to do would be to schedule an outage prior to the failover and only allow connections once the upgrade process is complete.

For this example, it will be moved to Node C. The reason for doing this is not apparent if you do not run the interface and just run a script where you can only interrogate the log file afterward. When you patch a clustered SQL Server, there are a series of validation checks that happen. There are three that matter the most:

- Cluster Group Owned by a Local Computer**  
 This checks to see if the node that is going to be updated owns any clustered SQL Server instances. Even if you do not intend on patching the instance that node is running, it will still generate a warning. Warnings are usually not problematic, but in this case, it is one that should be heeded. During the testing for this scenario, the assumption was made that even though I was not going to patch the instance owned by the node, the setup process would leave it alone and it could be manually moved after the installation successfully completed. That assumption was incorrect. What actually happens is that right before the installation completes, the running instance is failed over to another node automatically. So if you were allowing connections into it, they would be abruptly disconnected. This is clearly a scenario you want to avoid, so the move happens before the installation starts. Figure 5 shows the message when you click on the hyperlink for the warning.

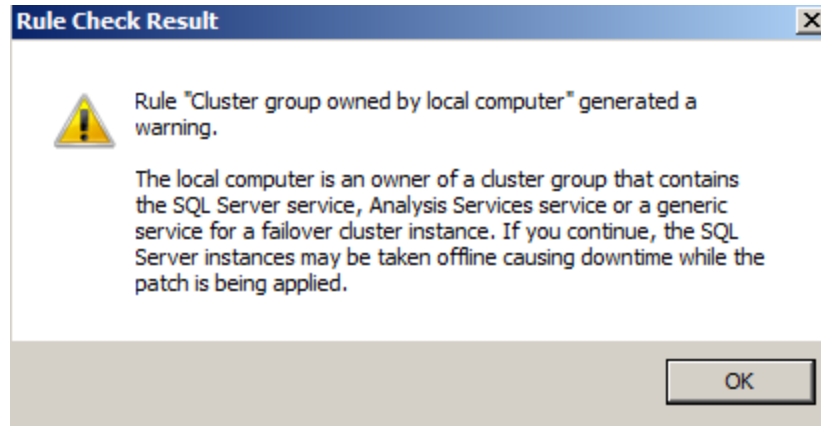


Figure 5. Detail of the ownership validation check

- **Cluster Resource DLL Update Restart Check**

If an instance of SQL Server is running on that node, the installation process also generates a warning because its resource DLL is in use. The resource DLL is fundamental to a clustered instance of SQL Server. Figure 6 shows the message when you click on the hyperlink for the warning.

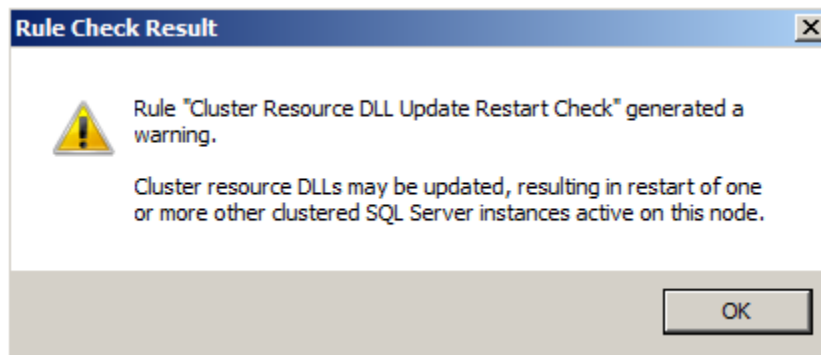


Figure 6. Detail of the cluster resource DLL check

- **Check Files In Use**

For SQL Server to be able to complete the installation of the binaries, the files it needs to replace cannot be in use. This is checked as part of the installation process. Figures 7 and 8 respectively show the results if no files are in use or if one or more files are in use. If a file is in use, the machine will definitely need a reboot. Do not assume that the file in use can just be stopped or unloaded, as it may be in use by other processes.

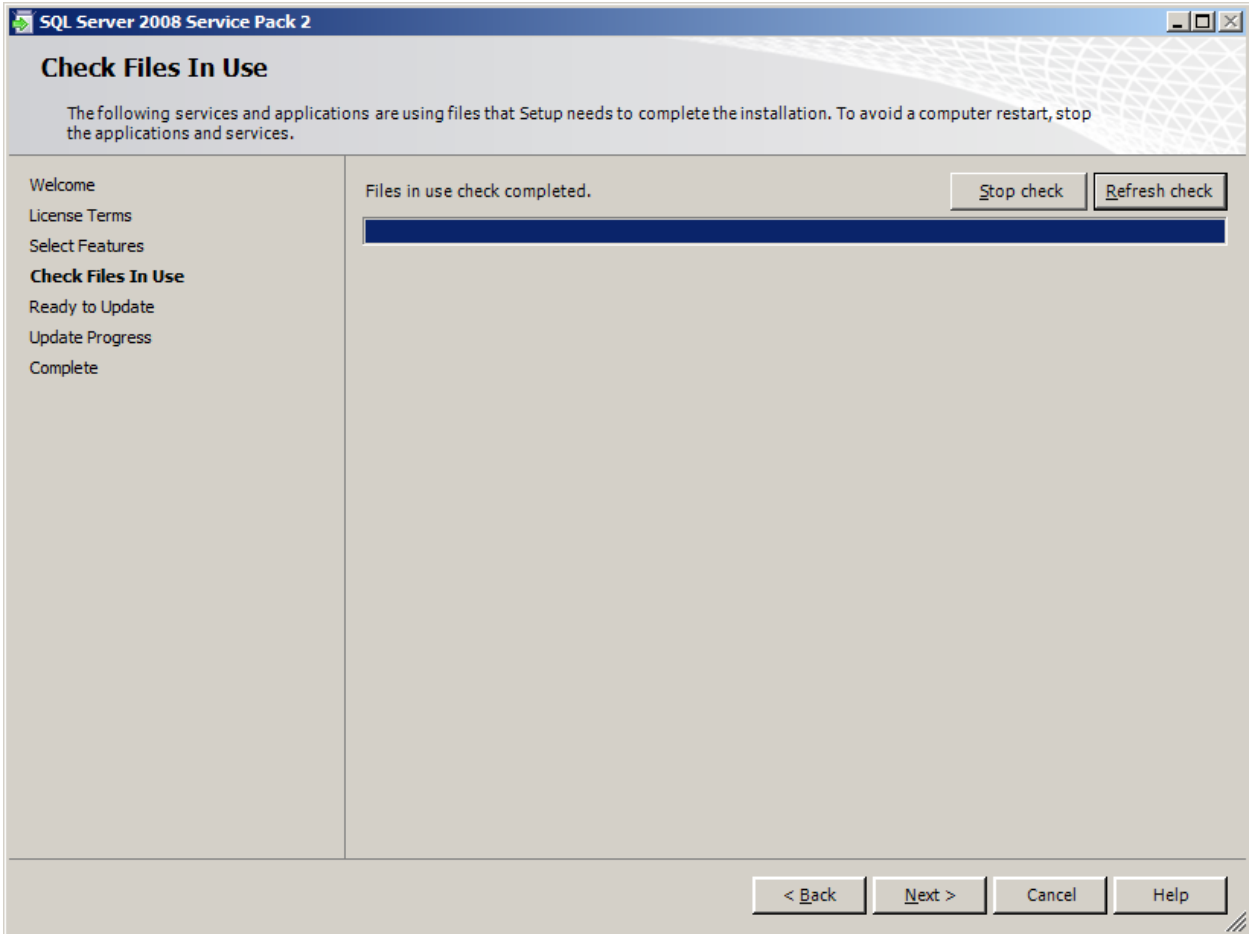


Figure 7. No files in use

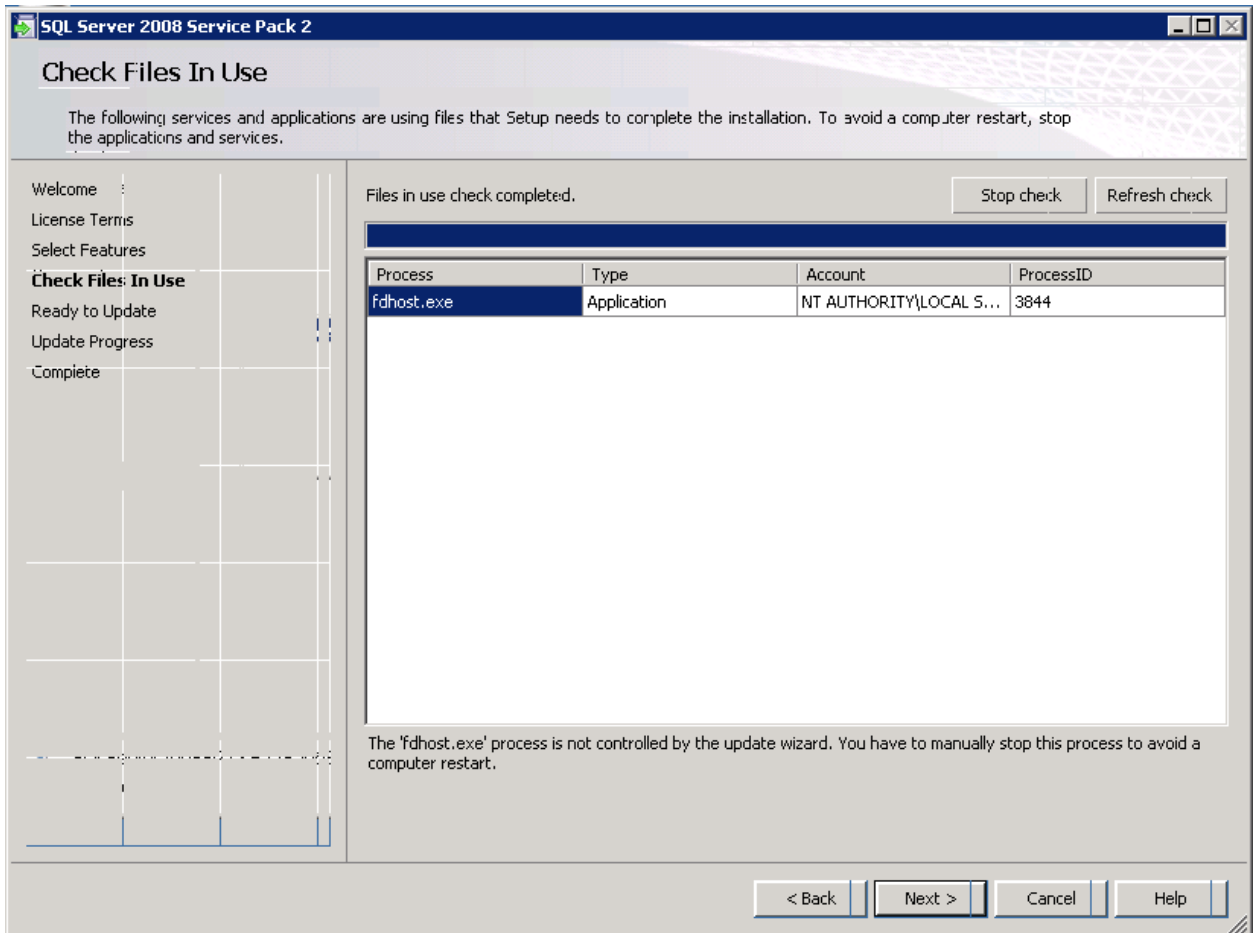
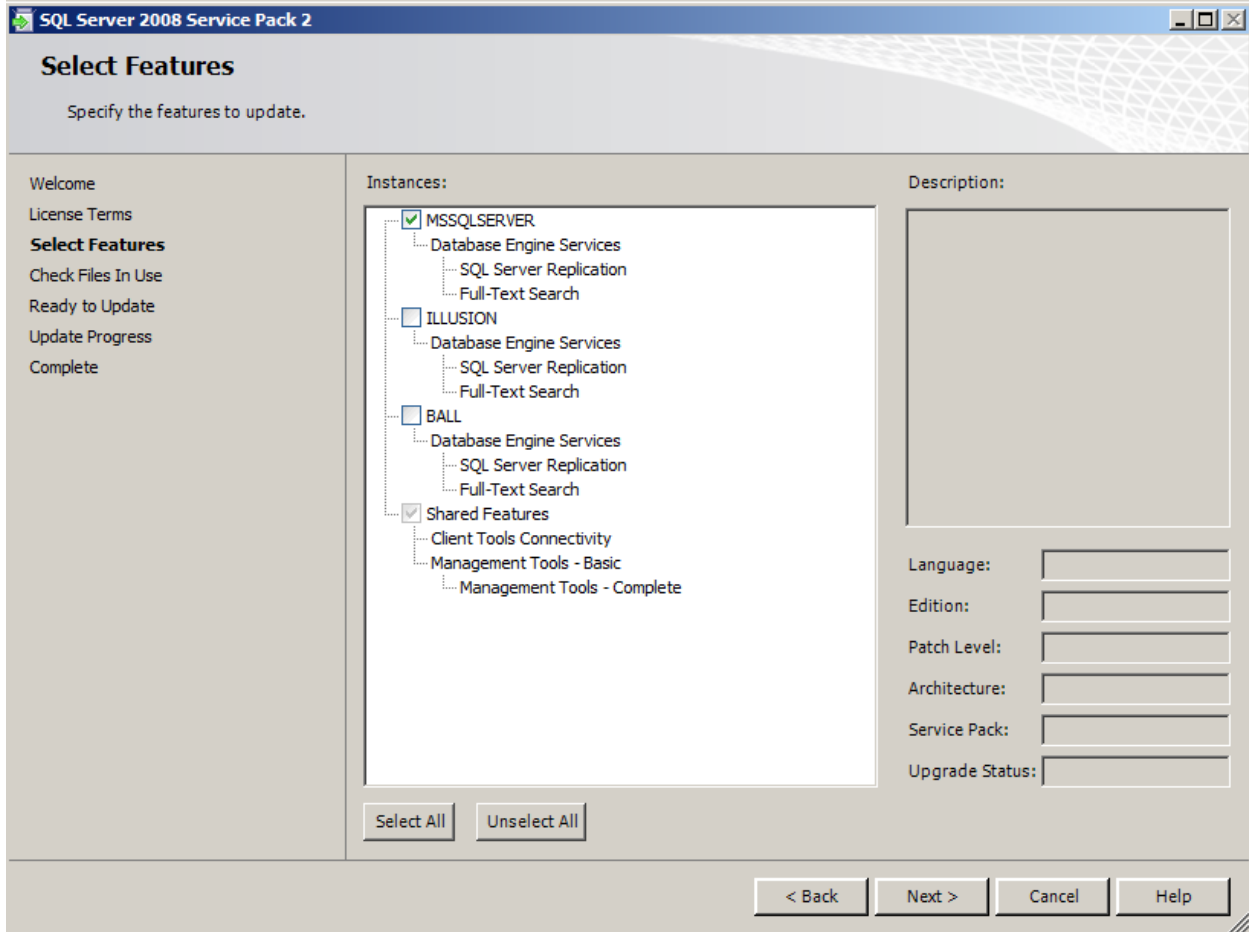


Figure 8. One file in use

	Node A			Node B			Node C		
	Owner	Possible Owner	Version	Owner	Possible Owner	Version	Owner	Possible Owner	Version
<b>Instance 1</b>	Yes	Yes	SP1	No	No	SP1	No	No	SP1
<b>Instance 2</b>	No	Yes	SP1	No	Yes	SP1	Yes	Yes	SP1
<b>Instance 3</b>	No	Yes	SP1	No	Yes	SP1	Yes	Yes	SP1

Table 3. State of the cluster after moving Instance 2

Run the SP2 install against Instance 1 on Node B. If you are not going to use scripts, on the Select Features dialog only select the instance that you are going to update. In Figure 9, an example is shown where only the default instance (MSSQLSERVER) is going to be updated since it has the checkmark next to it. Instance 1 and Instance 3 are still available to service requests.



**Figure 9. Selecting only one instance to update**

At this point, the configuration looks like the one shown in Table 4. The reason there is an asterisk next to SP1 is that on Node B, the shared files for all instances have been upgraded as part of the install of SP2 for Instance 1. Instance 2 and Instance 3 are still at SP1 (including their dedicated binaries). This scenario is completely supported by Microsoft and is a normal and expected part of the upgrade process.

	Node A			Node B			Node C		
	Owner	Possible Owner	Version	Owner	Possible Owner	Version	Owner	Possible Owner	Version
<b>Instance 1</b>	Yes	Yes	SP1	No	No	SP2	No	No	SP1
<b>Instance 2</b>	No	Yes	SP1	No	Yes	SP1*	Yes	Yes	SP1
<b>Instance 3</b>	No	Yes	SP1	No	Yes	SP1*	Yes	Yes	SP1

**Table 4. State after upgrading Instance 1 on Node 2**

You may or may not be prompted to reboot the node after installing SP2. If so, reboot Node B. In testing this scenario and a few others I tried, I found that sometimes it would prompt for a reboot and other times it would not even if the node did not own any instances. If you are installing from the GUI, you will see the dialog shown in Figure 9. Figure 10 shows the output if using the command line to install SP2.

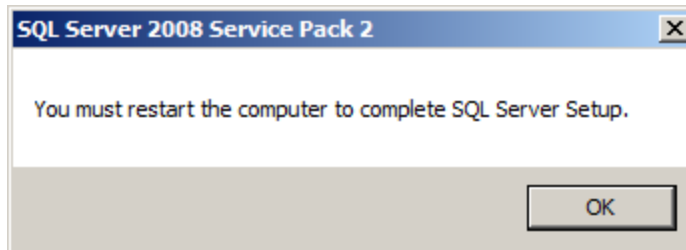


Figure 9. Dialog prompting for a reboot

```
C:\Install\Upgrade Scripts - Scenario 1>"C:\Install\SP2\setup.exe" /Action=patch
/instancename=MSSQLSERVER /quiet
Microsoft (R) SQL Server 2008 Setup 10.00.4000.00
Copyright (c) Microsoft Corporation. All rights reserved.
```

One or more affected files have operations pending.  
You should restart your computer to complete this process.

Figure 10. Reboot request from a command line installation

**TIP:** The upgrade scripts that correspond to this white paper only use the /QUIET parameter, which shows no GUI and no progress status during the installation. If you want to see the SP2 install output as it's running, add /INDICATEPROGRESS=TRUE to the script. You will see output similar to Figure 11 (which will constantly scroll and change with progress), which shows the beginning of the process. Figure 12 shows a reboot request with the verbose output.

```
C:\Install\Upgrade Scripts - Scenario 1>"C:\Install\SP2\setup.exe" /Action=patch
/instancename=ILLUSION /quiet /IndicateProgress=True
Microsoft (R) SQL Server 2008 Setup 10.00.4000.00
Copyright (c) Microsoft Corporation. All rights reserved.
```

```
Log provider 'Microsoft.SqlServer.Chainer.Infrastructure.LogProviderFile' has be
en registered
Log provider 'Microsoft.SqlServer.Chainer.Infrastructure.LogProviderConsole' has
been registered
Log provider 'Microsoft.SqlServer.Chainer.Infrastructure.LogProviderStatus' has
been registered
Reading XML resource 'Microsoft.SQL.Chainer.PackageData.Package.xml'
Reading schema resource 'Microsoft.SQL.Chainer.Package.Package.xsd'
```

Figure 11. Using /IndicateProgress=True

```
Running Action: CloseUI
Stop action skipped in UI Mode Quiet
Completed Action: CloseUI, returned True
One or more affected files have operations pending.
You should restart your computer to complete this process.
```

---

Setup result: 3010

Figure 12. Reboot request with verbose output

If you saw no visual prompt for a reboot (as shown earlier in Figures 10 and 12), use one of the following methods to check whether the upgrade was successful and doesn't need a reboot. Figure 13 shows the output of a successful verbose installation with no reboot message.

```
Running Action: CloseUI
Stop action skipped in UI Mode Quiet
Completed Action: CloseUI, returned True
```

---

Setup result: 0

Figure 13. Successful run with no reboot message

#### Method 1 – Check the Installation Summary File

1. Navigate to the `C:\Program Files\Microsoft SQL Server\100\Setup Bootstrap\Log` folder.
2. You will see a file named `Summary.txt`. Open it.
3. It will be evident, as seen in Figure 14, that the upgrade was successful but needs a reboot.

```
Overall summary:
Final result:           Passed but reboot required, see logs for details
Exit code (Decimal):   3010
Exit message:          Passed but reboot required, see logs for details
Start time:            2011-03-10 18:07:05
End time:               2011-03-10 18:21:11
Requested action:      Patch

Instance MSSQLSERVER overall summary:
Final result:           Passed but reboot required, see logs for details
Exit code (Decimal):   3010
Exit message:          Passed but reboot required, see logs for details
Start time:            2011-03-10 18:09:47
End time:               2011-03-10 18:20:57
Requested action:      Patch
```

Figure 14. Summary.txt showing a reboot is required

#### Method 2 – Check the Installation Detail Log File

1. Navigate to the `C:\Program Files\Microsoft SQL Server\100\Setup Bootstrap\Log` folder.
2. You will see a dated folder for each installation attempt. The most recent folder will be the SP2 installation in this scenario. In the example shown in Figure 15, `20110208_220710` is the latest directory. Open it.

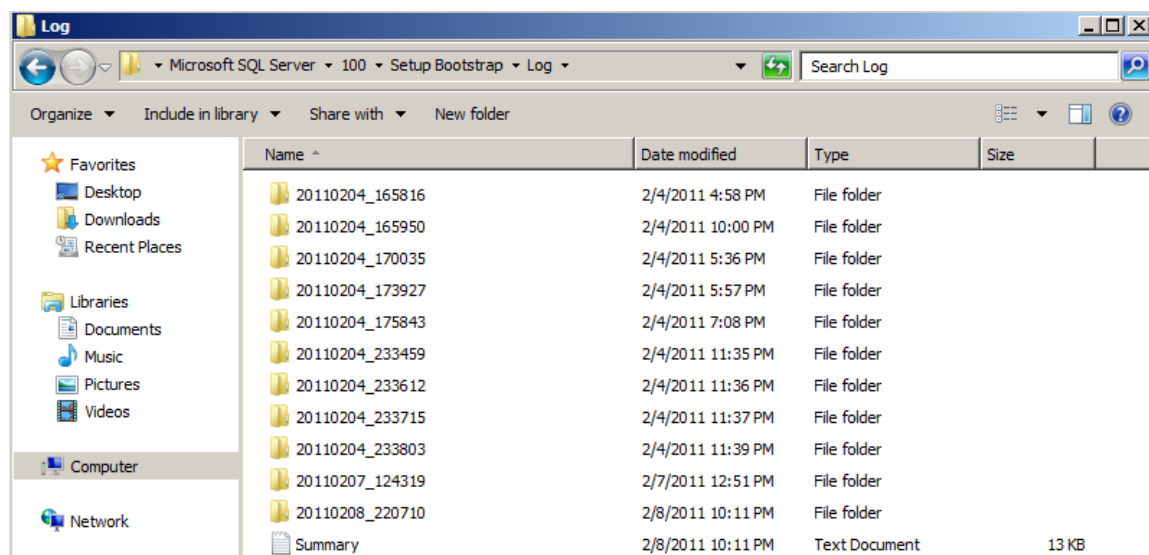


Figure 15. Sample log folder

3. In that dated folder, there will be a file called `Detail.txt`. Open it.
4. In `Detail.txt`, look for the following:
  - a. Search for the word "reboot" until you find the section where it checks whether a reboot is needed. It should clearly indicate a reboot is not necessary. You can also use the methods shown earlier. A sample is shown below in Figure 16.

```
2011-02-08 22:08:26 Slp: Init rule target object: Microsoft.SqlServer.Configuration.SetupExtension.FacetRebootRequired
2011-02-08 22:08:26 Slp: Rule 'RebootRequiredCheck' results: IsRebootNotRequired=True
2011-02-08 22:08:26 Slp: Evaluating rule : RebootRequiredCheck
2011-02-08 22:08:26 Slp: Rule running on machine: TOMMY
2011-02-08 22:08:26 Slp: Rule evaluation done : Succeeded
2011-02-08 22:08:26 Slp: Rule evaluation message: The computer does not require a restart.
```

Figure 16. Showing no reboot is required

- b. Scroll through the file. There should be no failures or errors. If you are patching more than one instance at a time, you will need to look for each instance. Figure 17 shows one of the samples you can look for to determine if the upgrade was a success.



```

2011-02-08 22:09:12 slp: Attempting to run patch request for instance: ILLUSION
2011-02-08 22:11:46 slp: Successfully ran patch request for instance: ILLUSION (exit code: 0)
2011-02-08 22:11:46 slp: Attempting to open SS Manager

```

Figure 17. Sample of showing success in the log

### Method 3 – Check the Registry

While checking the log file is always recommended, another place that will definitively show whether files are waiting to be changed and can only be performed using a reboot is in the registry. If *PendingFileRenameOperations* exists for the key HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager, you need to reboot that node. If it does not exist, then no files require a reboot. An example of where a reboot would be necessary is shown in Figure 18. If no reboot is required, PendingFileRenameOperations should not exist, as shown in Figure 19. Even if PendingFileRenameOperations does not exist, that doesn't necessarily mean a reboot isn't required; the log file for the installation is the final word.

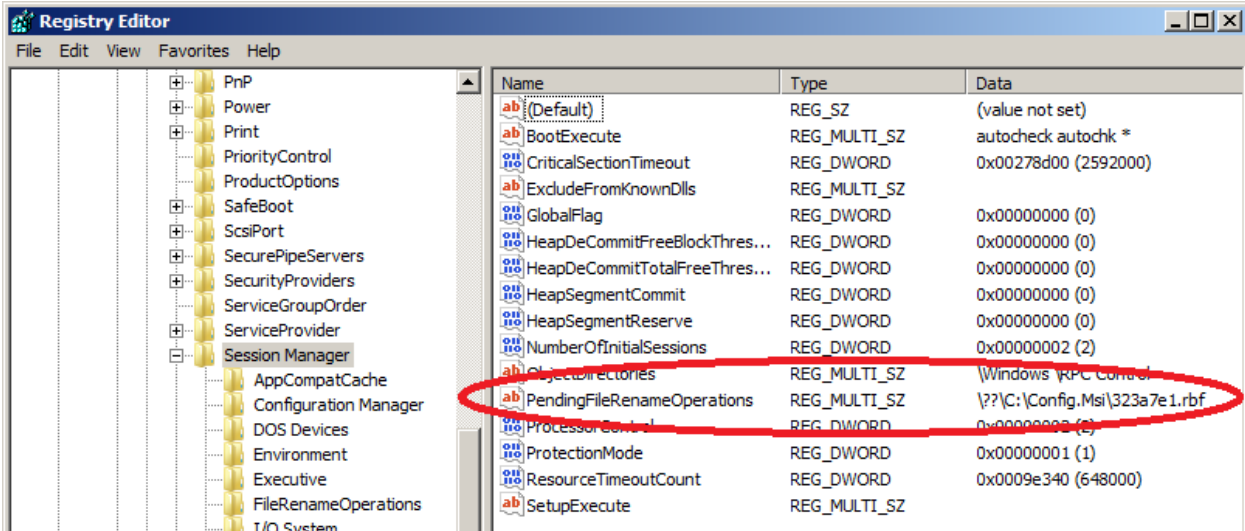


Figure 18. Session Manager with PendingFileRenameOperations

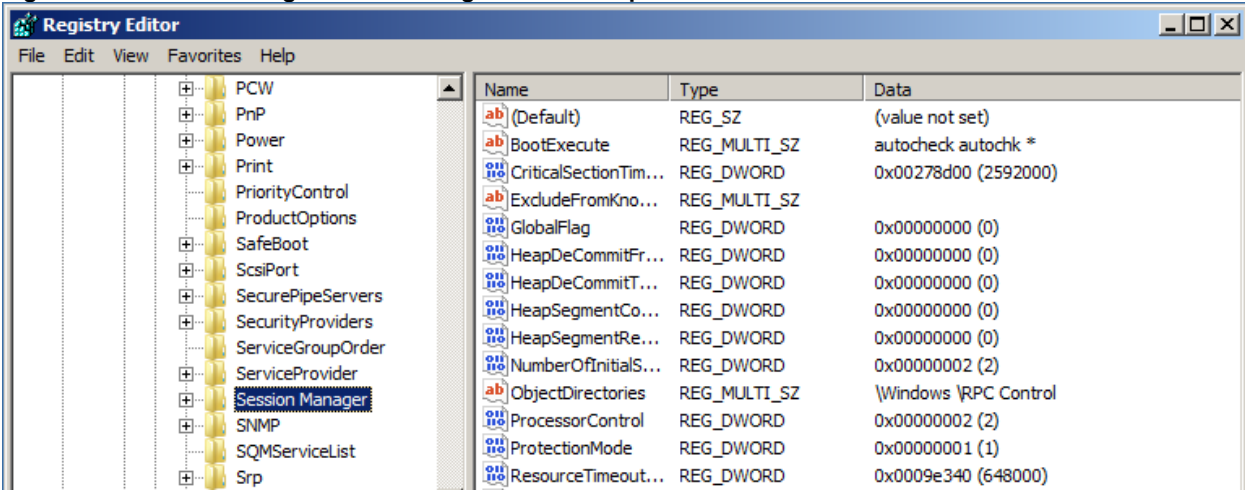


Figure 19. Session Manager without PendingFileRenameOperations

**TIP:** The registry can also be checked via PowerShell. A script named *PFRCheck.ps1* is included in the zip file. If a reboot is necessary, you will see the message: "PendingFileRenameOperations does exist and has values. You must reboot this server." If a reboot is not necessary, you will see this message: "PendingFileRenameOperations does not exist or has no values. No reboot is required."

To avoid the automatic failover of an instance that would happen if the node owned the resources, to facilitate the upgrade of Instance 1 on Node C, fail both Instance 2 and Instance 3 to Node B. The status is shown in Table 5. Similar to Instance 2, because Instance 3 will be impacted by failovers or reboots, you should schedule an outage until the upgrade process is complete for that instance.

S

	Node A			Node B			Node C		
	Owner	Possible Owner	Version	Owner	Possible Owner	Version	Owner	Possible Owner	Version
<b>Instance 1</b>	Yes	Yes	SP1	No	No	SP2	No	No	SP1
<b>Instance 2</b>	No	Yes	SP1	Yes	Yes	SP1*	No	Yes	SP1
<b>Instance 3</b>	No	Yes	SP1	Yes	Yes	SP1*	No	Yes	SP1

**Table 5. Moving Instances 2 and 3 to Node B**

Now that Node C owns no instances, upgrade Instance 1. As with Node B, you may or may not need a reboot. The cluster will look like Table 6.

	Node A			Node B			Node C		
	Owner	Possible Owner	Version	Owner	Possible Owner	Version	Owner	Possible Owner	Version
<b>Instance 1</b>	Yes	Yes	SP1	No	No	SP2	No	No	SP2
<b>Instance 2</b>	No	Yes	SP1	Yes	Yes	SP1*	No	Yes	SP1*
<b>Instance 3</b>	No	Yes	SP1	Yes	Yes	SP1*	No	Yes	SP1*

**Table 6. State after upgrading Instance 1 on Node 3**

Add Node B and Node C back as possible owners of Instance 1 as shown in Table 7.

	Node A			Node B			Node C		
	Owner	Possible Owner	Version	Owner	Possible Owner	Version	Owner	Possible Owner	Version
<b>Instance 1</b>	Yes	Yes	SP1	No	Yes	SP2	No	Yes	SP2
<b>Instance 2</b>	No	Yes	SP1	Yes	Yes	SP1*	No	Yes	SP1*
<b>Instance 3</b>	No	Yes	SP1	Yes	Yes	SP1*	No	Yes	SP1*

**Table 7. Instance 1 owned by all nodes**

Once Nodes B and C can own Instance 1, you have to fail it over to one of them to start the upgrade of the instance itself. In this example scenario, Instance 1 will be moved to Node B as shown in Table 8. Schedule an outage and fail it over. If you try to connect to the instance during its upgrade with a tool like SQL Server Management Studio, you will see the dialog in Figure 20.

	Node A			Node B			Node C		
	Owner	Possible Owner	Version	Owner	Possible Owner	Version	Owner	Possible Owner	Version
Instance 1	No	Yes	SP1	Yes	Yes	SP2	No	Yes	SP2
Instance 2	No	Yes	SP1	Yes	Yes	SP1*	No	Yes	SP1*
Instance 3	No	Yes	SP1	Yes	Yes	SP1*	No	Yes	SP1*

Table 8. Moving Instance 1 to Node B to start its upgrade

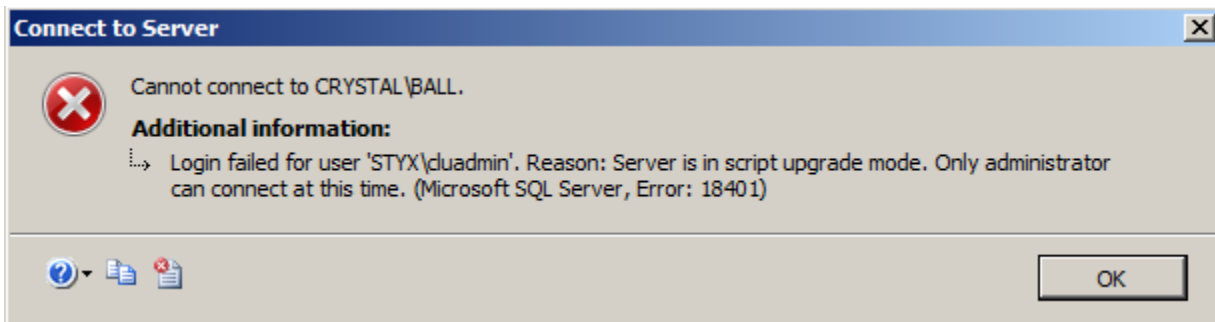


Figure 20. Message letting you know the instance upgrade is in progress

Once Instance 1 is upgraded, verify that it is at SP2. There are a few ways you can do this. Three of the easiest are:

- Start SQL Server Management Studio and connect to the instance. It should reflect the new version number. Figure 21 shows the instance EQUINOX has been upgraded to SP2, but CRYSTAL\BALL is still at SP1.
  - CRYSTAL\BALL (SQL Server 10.0.2531 - STYX\cluadmin)
    - Databases
    - Security
    - Server Objects
    - Replication
    - Management
    - SQL Server Agent
  - EQUINOX (SQL Server 10.0.4000 - STYX\cluadmin)
    - Databases
    - Security
    - Server Objects
    - Replication
    - Management
    - SQL Server Agent

Figure 21. Output showing the instance is at the SP2 with the appropriate build number

- Run the query `SELECT @@VERSION`. An example is shown in Figure 22.

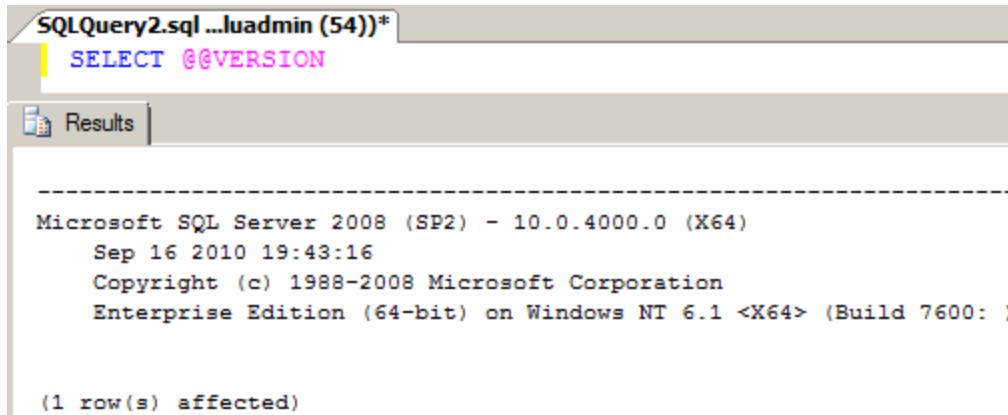


Figure 22. Output showing the instance is at the SP2 build number

- Run the query `SELECT SERVERPROPERTY ('ProductLevel')`, which should show the update level you just installed. An example is shown in Figure 23.

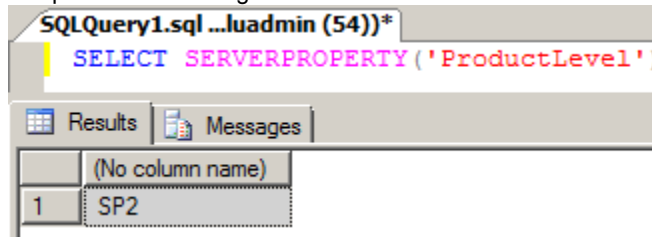


Figure 23. Output showing the instance is at SP2

Technically Instance 1 could be opened up for use since it can live in either Node B or C, but you have not finished the upgrade on Node A and, therefore, all of its failover tests. What you may want to do is run and complete any application-level tests you would normally execute against it to verify that the upgrade did not introduce anything unintended. You should have tested this prior to the production upgrade, but this round would be a shorter “smokescreen” set of tests against the most used functionality.

All three instances are owned by Node B. If you did not design your nodes to handle this situation, it could be a potential problem. With Instance 1, technically it could go to Node C but you still have two instances to update there and all three on Node A. The choice was made to run the upgrades on A and C to complete them. Therefore Node A must be removed as a possible owner from Instance 1, and Nodes A and C removed as possible owners of Instances 2 and 3. The state of the cluster is shown in Table 9.

One thing that should be pointed out is that Instances 2 and 3, like Instance 1 earlier, have nowhere to go. If they encounter a problem while the binaries are upgraded on Nodes A or C, they cannot fail over anywhere else. This should be deemed an acceptable risk and cannot be avoided. At some point in this scenario, it is impossible to avoid an instance being stranded somewhere. The second example scenario shows how this situation could be avoided.

	Node A			Node B			Node C		
	Owner	Possible Owner	Version	Owner	Possible Owner	Version	Owner	Possible Owner	Version
Instance 1	No	No	SP1	Yes	Yes	SP2	No	Yes	SP2
Instance 2	No	No	SP1	Yes	Yes	SP1*	No	No	SP1*
Instance 3	No	No	SP1	Yes	Yes	SP1*	No	No	SP1*

Table 9. State after upgrading Instance 1 on Nodes 2 and 3

Run the upgrade of SP2 on Node A against all three instances and on Node C against Instance 2 and Instance 3. The upgrade processes can be run in parallel or just sequentially since the operations are on two different nodes. If you are risk adverse, only do one node at a time. If you are using the GUI, Figure 24 shows an example of what will happen on Node C. The GRANDVILLUSION (showing as ILLUSION) and CRYSTALBALL (showing as BALL) instances can be upgraded, but notice that the default (MSSQLSERVER) instance that was already upgraded cannot be selected.

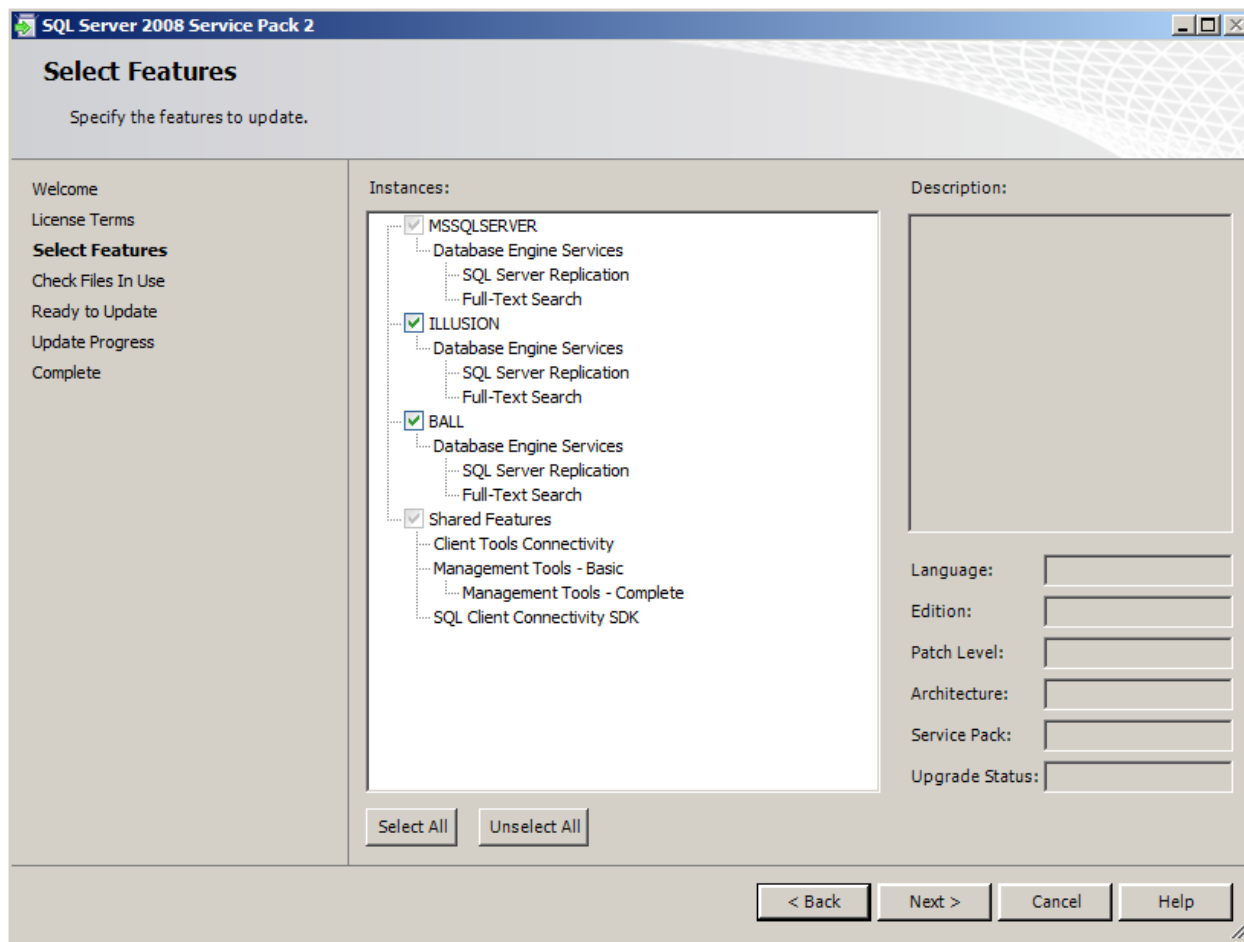


Figure 24. Upgrading two instances when others have already been upgraded

Table 10 shows the state of the cluster after running the upgrades on Nodes A and C.

	Node A			Node B			Node C		
	Owner	Possible Owner	Version	Owner	Possible Owner	Version	Owner	Possible Owner	Version
Instance 1	No	No	SP2	Yes	Yes	SP2	No	Yes	SP2
Instance 2	No	No	SP2	Yes	Yes	SP1*	No	No	SP2
Instance 3	No	No	SP2	Yes	Yes	SP1*	No	No	SP2

Table 10. State after upgrading multiple instances on Nodes A and C

Add Node A back as a possible owner of Instance 1 and Nodes A and C back as possible owners of Instance 2 and Instance 3. The cluster will be in the state shown in Table 11. You may or may not need to reboot those nodes. In this

example, it is a three node cluster, so rebooting two nodes at the same time would cause the failover cluster to lose quorum because it is using a Node Majority quorum model; a loss of two nodes pushes it over the limit for staying up. Reboot them one at a time.

	Node A			Node B			Node C		
	Owner	Possible Owner	Version	Owner	Possible Owner	Version	Owner	Possible Owner	Version
Instance 1	No	Yes	SP2	Yes	Yes	SP2	No	Yes	SP2
Instance 2	No	Yes	SP2	Yes	Yes	SP1*	No	Yes	SP2
Instance 3	No	Yes	SP2	Yes	Yes	SP1*	No	Yes	SP2

**Table 11. Changing the possible owners for multiple instances back on Nodes A and C**

For Instance 1, the only task left is to finish its cycle of failovers. Fail it over to Node C. Both Instance 2 and Instance 3 are still at SP1, so Node B must be upgraded in addition to the instances themselves. To upgrade the instances and facilitate upgrading the binaries on Node B, fail Instance 2 and Instance 3 to another node. In this example as shown in Table 12, that node is Node A.

	Node A			Node B			Node C		
	Owner	Possible Owner	Version	Owner	Possible Owner	Version	Owner	Possible Owner	Version
Instance 1	No	Yes	SP2	No	Yes	SP2	Yes	Yes	SP2
Instance 2	Yes	Yes	SP2	No	Yes	SP1*	No	Yes	SP2
Instance 3	Yes	Yes	SP2	No	Yes	SP1*	No	Yes	SP2

**Table 12. Moving instances to upgrade Node B and start the completion of the Instance 1 upgrade**

Remove Node B as a possible owner of Instances 2 and 3. The cluster state is shown in Table 13.

	Node A			Node B			Node C		
	Owner	Possible Owner	Version	Owner	Possible Owner	Version	Owner	Possible Owner	Version
Instance 1	No	Yes	SP2	No	Yes	SP2	Yes	Yes	SP2
Instance 2	Yes	Yes	SP2	No	No	SP1*	No	Yes	SP2
Instance 3	Yes	Yes	SP2	No	No	SP1*	No	Yes	SP2

**Table 13. Removing Node B as an owner of Instances 2 and 3**

Perform the last set of installations by running the SP2 install against Instances 2 and 3 on Node B. When the installation is complete, the cluster will look like Table 14.

	Node A			Node B			Node C		
	Owner	Possible Owner	Version	Owner	Possible Owner	Version	Owner	Possible Owner	Version
Instance 1	No	Yes	SP2	No	Yes	SP2	Yes	Yes	SP2
Instance 2	Yes	Yes	SP2	No	No	SP2	No	Yes	SP2
Instance 3	Yes	Yes	SP2	No	No	SP2	No	Yes	SP2

**Table 14. After upgrading two instances on Node B**

Add Node B back as a possible owner of Instance 2 and Instance 3 as shown in Table 15.

	Node A			Node B			Node C		
	Owner	Possible Owner	Version	Owner	Possible Owner	Version	Owner	Possible Owner	Version
Instance 1	No	Yes	SP2	No	Yes	SP2	Yes	Yes	SP2
Instance 2	Yes	Yes	SP2	No	Yes	SP2	No	Yes	SP2
Instance 3	Yes	Yes	SP2	No	Yes	SP2	No	Yes	SP2

**Table 15. Node B can now own all instances**

To finish the set of failovers and complete the upgrade for Instance 1, move it from Node C to Node A (its original owner). You could do this step earlier if you wanted to. For example, after you failed it over to C you could immediately move it back to A. For clarity's sake, it was done later in this example. Once the failover to Node A is successful, Instance A's upgrade process is finished.

To start the process of the failover tests for Instance 2 and Instance 3, fail Instance 2 to Node C and Instance 3 to Node B. The reason this order was selected was because the intent of this example is to have the instances end up where they started. Assuming you have the proper capacity, this does not have to be the case, but it is often what most want to have happen. The cluster state is shown in Table 16.

	Node A			Node B			Node C		
	Owner	Possible Owner	Version	Owner	Possible Owner	Version	Owner	Possible Owner	Version
Instance 1	Yes	Yes	SP2	No	Yes	SP2	No	Yes	SP2
Instance 2	No	Yes	SP2	No	Yes	SP2	Yes	Yes	SP2
Instance 3	No	Yes	SP2	Yes	Yes	SP2	No	Yes	SP2

**Table 16. Moving instances to other nodes in the cluster**

There is only one more series of failovers. Fail Instance 2 to Node B and Instance 3 to Node C. Assuming they were successful, all three instances have been successfully upgraded to SP2 and you know that all nodes can own the resources in a failover situation. As mentioned earlier, make sure that Instance 2 and Instance 3 are tested by their respective applications to ensure that everything is all right before allowing everyone to start using them.

	Node A			Node B			Node C		
	Owner	Possible Owner	Version	Owner	Possible Owner	Version	Owner	Possible Owner	Version
<b>Instance 1</b>	Yes	Yes	SP2	No	Yes	SP2	No	Yes	SP2
<b>Instance 2</b>	No	Yes	SP2	<i>Yes</i>	Yes	SP2	<i>No</i>	Yes	SP2
<b>Instance 3</b>	No	Yes	SP2	<i>No</i>	Yes	SP2	<i>Yes</i>	Yes	SP2

**Table 17. Final state after all failover tests**



# Detailed Steps for Scenario One

The steps below represent exactly what was done to complete this scenario. Some of the steps are combined in the scripts available for download as you would do in real life. The operating system used for this test was Windows Server 2008 R2, thus PowerShell was used for the Windows failover clustering commands.

A file with an extension of .ps1 denotes a PowerShell script, and a .bat file is a standard batch file that can be run in a standard command prompt window. In the corresponding scripts, the generic names map to the following:

Generic Name	Script Name
Instance 1	EQUINOX (resource group for the default instance), MSSQLSERVER (SP install)
Instance 2	CRYSTALBALL (resource group for the named instance of the same name)
Instance 3	GRAND\ILLUSION (resource group for the named instance of the same name)
Node A	DENNIS
Node B	TOMMY
Node C	JY

**NOTE:** Where a script is listed for multiple steps, you do not need to re-execute the script. The script runs the commands in sequence to automate things. Each step is broken out for clarity here.

Step	Details	Node	Script to Execute
1	Remove Node B and Node C as possible owners of Instance 1's network name resource.	Any	1.ps1
2	Stop all application and user connections into Instance 2.	N/A	N/A
3	Fail Instance 2 to JY.	Any	2.ps1
4	Run the SP2 install against Instance 1 on Node B.	Node B	InstallSPDefault.bat
5	Reboot Node B if necessary.	Node B	Reboot.ps1 (can be modified to run from other nodes if the person executing has the right permissions)
6	When reboot is complete, fail Instance 2 back to Node B.	Any	3.ps1
7	Stop all application and user connections into Instance 3.	N/A	N/A
8	Fail Instance 3 to Node B.	Any	4.ps1
9	Run the SP2 install against Instance 1 on Node C.	Node C	InstallSPDefault.bat
10	Reboot Node C if necessary.	Node C	Reboot.ps1 (can be modified to run from other nodes if the person executing has the right permissions)
11	Add Node B and Node C back as possible owners of Instance 1's network name resource.	Any	5.ps1
12	Stop all application and user connections into Instance 1.	N/A	N/A
13	Fail Instance 1 to Node B.	Any	6.ps1
14	Verify Instance 1 is now at SP2. While it can technically be used for testing or production, but not all failover tests have been completed, so an additional outage would need to be accounted for.	Any	Use your preferred method
15	Remove Node A as a possible owner of Instance	Any	7.ps1

	1's network name resource.		
16	Remove Node A and Node C as possible owners of Instance 2's network name resource.	Any	7.ps1
17	Remove Node A and Node C as possible owners of Instance 3's network name resource.	Any	7.ps1
18	Run the SP2 install against Instance 1, Instance 2, and Instance 3.	Node A	InstallSPAll.bat
19	Reboot Node C if necessary.	Node A	Reboot.ps1 (can be modified to run from other nodes if the person executing has the right permissions)
20	Run the SP2 install against Instance 2 and Instance 3.	Node C	InstallSPAll.bat
21	Reboot Node C if necessary.	Node C	Reboot.ps1 (can be modified to run from other nodes if the person executing has the right permissions)
22	Add Node A back as possible owner of Instance 1's network name resource.	Any	8.ps1
23	Add Node A and Node C back as possible owners of Instance 2's network name resource.	Any	8.ps1
24	Add Node A and Node C back as possible owners of Instance 3's network name resource.	Any	8.ps1
25	Fail Instance 1 to Node C.	Any	9.ps1
26	Fail Instance 2 to Node A.	Any	10.ps1
27	Fail Instance 3 to Node A.	Any	11.ps1
28	Verify Instance 2 is now at SP2. While it can technically be used for testing or production, but not all failover tests have been completed, so an additional outage would need to be accounted for.	Any	Use your preferred method
29	Verify Instance 3 is now at SP2. While it can technically be used for testing or production, but not all failover tests have been completed, so an additional outage would need to be accounted for.	Any	Use your preferred method
30	Remove Node B as a possible owner of Instance 2's network name resource.	Any	12.ps1
31	Remove Node B as a possible owner of Instance 3's network name resource.	Any	12.ps1
32	Run the SP2 install against Instance 2 and Instance 3.	Node B	InstallSPAll.bat
33	Add Node B back as a possible owner of Instance 2's network name resource.	Any	13.ps1
342	Add Node B back as a possible owner of Instance 3's network name resource.	Any	13.ps1
35	Fail Instance 2 to Node C.	Any	14.ps1
36	Fail Instance 3 to Node B.	Any	15.ps1
37	Fail Instance 1 to Node A. This will complete failover testing for Instance 1 and assuming no problems, it can be opened for other testing and/or production usage.	Any	16.ps1
38	Fail Instance 2 to Node B. This will complete failover testing for Instance 2 and assuming no problems, it can be opened for other testing and/or production usage.	Any	17.ps1
39	Fail Instance 3 to Node C. This will complete failover testing for Instance 3 and assuming no problems, it can be opened for other testing and/or production usage.	Any	18.ps1

# Conclusion for Scenario One

---

While not quite an instance-by-instance upgrade, it is very close. You could take some different paths than the ones shown here (such as starting with different nodes or changing where to fail things over to), but not much would change and the total number of steps would be about the same. As you can see, this method would require you to carefully negotiate downtime since there could be reboots in addition to the multiple failovers necessary.

Some instances may be able to be up longer than others. Due to the fact that upgrading an instance that is not running on a node while another is running will cause an automatic failover of the running instance, navigating is tricky to ensure that circumstance does not happen. This means that all of the nodes will have various outages, so while the actual upgrade time for the instance is very short once it has somewhere to go, getting there and completing everything is where most of the downtime will occur.

This scenario highlights how difficult the patching process can be when you don't have any reserve space to move instances to in the cluster. While you should be able to complete this entire process in a few hours, the impact on the applications and users is something that will have to be negotiated and carefully coordinated, especially if you only have a limited window of time to perform the upgrade. When your cluster was originally designed, you may not have planned for an additional, dedicated failover node — or had the money for it. But not having one could prove far more costly down the road, as this scenario demonstrates.

# Scenario Two – Dedicated Failover Node

This second scenario will show you how to apply SQL Server 2008 SP2 to a cluster that has three instances spread out over four nodes. One node (Node D) is considered a dedicated failover node. Every other node has one instance running on it. The initial state is shown in Table 1. The way to read the tables remains the same as the previous scenario.

	Node A			Node B			Node C			Node D		
	Owner	Possible Owner	Version	Owner	Possible Owner	Version	Owner	Possible Owner	Version	Owner	Possible Owner	Version
<b>Instance 1</b>	Yes	Yes	SP1	No	Yes	SP1	No	Yes	SP1	No	Yes	SP1
<b>Instance 2</b>	No	Yes	SP1	Yes	Yes	SP1	No	Yes	SP1	No	Yes	SP1
<b>Instance 3</b>	No	Yes	SP1	No	Yes	SP1	Yes	Yes	SP1	No	Yes	SP1

**Table 1. Initial cluster state**

Because there is a dedicated failover node, and each node has a single instance, removing half the nodes is much easier. Only one instance would need to be failed over to another node initially. In this example, Instance 3 is moved to Node B. This leaves both Node C and Node D owning no instances. This means they can be removed as possible owners of the network name resource for all three instances and patched at the same time. There is no instance-by-instance shuffle as there was in the previous scenario. The state of the cluster pre-upgrade should look like Table 2.

You'll need to schedule a quick outage to move Instance 3 to Node B. But unlike the first scenario, you can probably let users and applications still use it after the move to Node B. The reasons will become apparent as you read further.

	Node A			Node B			Node C			Node D		
	Owner	Possible Owner	Version	Owner	Possible Owner	Version	Owner	Possible Owner	Version	Owner	Possible Owner	Version
<b>Instance 1</b>	Yes	Yes	SP1	No	Yes	SP1	No	No	SP1	No	No	SP1
<b>Instance 2</b>	No	Yes	SP1	Yes	Yes	SP1	No	No	SP1	No	No	SP1
<b>Instance 3</b>	No	Yes	SP1	Yes	Yes	SP1	No	No	SP1	No	No	SP1

**Table 2. State of the cluster prior to any instance upgrades**

Execute the installation of SP2 on both Node C and Node D against all the instances (1, 2 and 3). No reboot should be required, but double-check using the methods shown in the last scenario. If reboots are needed, do so now. No instances will incur downtime for a reboot, because Nodes C and D do not own any. Remember that during this time the only outage incurred has been the brief scheduled one to Instance 3. The state of the cluster at this point is reflected in Table 3.

	Node A			Node B			Node C			Node D		
	Owner	Possible Owner	Version	Owner	Possible Owner	Version	Owner	Possible Owner	Version	Owner	Possible Owner	Version
<b>Instance 1</b>	Yes	Yes	SP1	No	Yes	SP1	No	No	SP2	No	No	SP2
<b>Instance 2</b>	No	Yes	SP1	Yes	Yes	SP1	No	No	SP2	No	No	SP2
<b>Instance 3</b>	No	Yes	SP1	Yes	Yes	SP1	No	No	SP2	No	No	SP2

**Table 3. State of the cluster after Node 3 and Node 4 upgrades**

At this point, the other instances will be affected. In this scenario, a “big bang” approach has been taken. Instead of now applying SP2 in a more individual way, all instances will be affected to minimize the overall downtime. Nodes C and D are added back as possible network name owners for all the instances. Instance 1, Instance 2, and Instance 3 are moved to either Node C or Node D. As shown in the previous scenario, once the instances are failed over to the nodes that have been upgraded to SP2, they will be upgraded from SP1 to SP2. Node 1 and Node 2 should be removed as possible owners of the network name for all three instances. The cluster state is shown in Table 4.

While the instances can technically be used for testing applications to ensure they still behave properly after the upgrade or even for production use right now, it is not recommended because all of the instances have not had their full failover tests and an additional outage will be required.

	Node A			Node B			Node C			Node D		
	Owner	Possible Owner	Version	Owner	Possible Owner	Version	Owner	Possible Owner	Version	Owner	Possible Owner	Version
<b>Instance 1</b>	No	No	SP1	No	No	SP1	Yes	Yes	SP2	No	Yes	SP2
<b>Instance 2</b>	No	No	SP1	No	No	SP1	Yes	Yes	SP2	No	Yes	SP2
<b>Instance 3</b>	No	No	SP1	No	No	SP1	No	Yes	SP2	Yes	Yes	SP2

**Table 4. State of the cluster after failovers to upgraded nodes**

The SP2 install should be run for all instances on both Node A and Node B. No reboot should be required, but double-check using the methods shown in the last scenario. If reboots are needed, do so now. No instances will incur downtime for a reboot because Nodes A and B do not own any. The cluster state is shown in Table 5.

	Node A			Node B			Node C			Node D		
	Owner	Possible Owner	Version	Owner	Possible Owner	Version	Owner	Possible Owner	Version	Owner	Possible Owner	Version
<b>Instance 1</b>	No	No	SP2	No	No	SP2	Yes	Yes	SP2	No	Yes	SP2
<b>Instance 2</b>	No	No	SP2	No	No	SP2	Yes	Yes	SP2	No	Yes	SP2
<b>Instance 3</b>	No	No	SP2	No	No	SP2	No	Yes	SP2	Yes	Yes	SP2

**Table 5. State of the cluster after upgrading Node 1 and Node 2**

Nodes A and B can be added back as possible owners of the network name resource for all three instances. At this point, only one major task remains: to test failover of the upgraded instances to all nodes. Assuming all upgrades and failovers were successful, the process to upgrade the instances from SP1 to SP2 is complete. The final cluster state is shown in Table 6.

	Node A			Node B			Node C			Node D		
	Owner	Possible Owner	Version	Owner	Possible Owner	Version	Owner	Possible Owner	Version	Owner	Possible Owner	Version
<b>Instance 1</b>	Yes	Yes	SP2	No	Yes	SP2	Yes	Yes	SP2	No	Yes	SP2
<b>Instance 2</b>	No	Yes	SP2	Yes	Yes	SP2	Yes	Yes	SP2	No	Yes	SP2
<b>Instance 3</b>	No	Yes	SP2	No	Yes	SP2	Yes	Yes	SP2	No	Yes	SP2

**Table 6. Completed upgrade to SQL Server 2008 SP2**

# Detailed Steps for Scenario Two

The steps below represent exactly what was done to complete this scenario. Some of the steps are combined in the scripts available for download as you would do in real life. The operating system used for this test was Windows Server 2008 R2, thus PowerShell was used for the Windows failover clustering commands.

A file with an extension of .ps1 denotes a PowerShell script, and a .bat file is a standard batch file that can be run in a standard command prompt window. In the corresponding scripts, the generic names map to the following:

Generic Name	Script Name
Instance 1	EQUINOX
Instance 2	CRYSTALBALL
Instance 3	GRANDILLUSION
Node A	DENNIS
Node B	TOMMY
Node C	JY
Node D	JOHN

**NOTE:** Where a script is listed for multiple steps, you do not need to re-execute the script. The script runs the commands in sequence to automate things. Each step is broken out for clarity here.

Step	Details	Node	Task/Script to Execute
1	Stop all application and user connections into Instance 3.	N/A	N/A
2	Fail Instance 3 to Node B. This should be a brief outage and production use can continue after the failover.	Any	1.ps1
3	Remove Node 3 and Node 4 as possible owners of the network name resource for Instance 1, Instance 2, and Instance 3.	Any	2.ps1
4	Upgrade all instances.	Node 3, Node 4	InstallSPAll.bat
5	Add Node 3 and Node 4 as possible owners of the network name resource for Instance 1, Instance 2, and Instance 3.	Any	3.ps1
6	Stop all application and user connections for all three instances to allow the rest of the patching process to continue.	N/A	N/A
7	Fail Instance 1 to Node 3.	Any	4.ps1
8	Fail Instance 2 to Node 3.	Any	4.ps1
9	Fail Instance 3 to Node 4.	Any	4.ps1
10	Remove Node 1 and Node 2 as possible owners of the network name resource for Instance 1, Instance 2, and Instance 3.	Any	5.ps1
11	Verify Instance 1 is now at SP2. It can technically be used for testing or production, but not all failover tests have been completed, so an additional outage would need to be accounted for.	Any	Use your preferred method
12	Verify Instance 2 is now at SP2. It can technically be used for testing or production, but not all failover tests have been completed, so an additional outage would need to be accounted for.	Any	Use your preferred method
13	Verify Instance 3 is now at SP2. It can technically be used for testing or production, but not all failover tests have been completed, so an additional outage would need to be accounted for.	Any	Use your preferred method
14	Upgrade all instances.	Node 1, Node 2	InstallSPAll.bat

15	Add Node 1 and Node 2 as possible owners of the network name resource for Instance 1, Instance 2, and Instance 3.	Any	3.ps1
16	Fail Instance 2 to Node 4.	Any	6.ps1
17	Fail Instance 2 to Node 1.	Any	6.ps1
18	Fail Instance 2 to Node 2. This will complete failover testing for Instance 2 and assuming no problems, it can be opened for other testing and/or production usage.	Any	6.ps1
19	Fail Instance 1 to Node 4.	Any	6.ps1
20	Fail Instance 1 to Node 2.	Any	6.ps1
21	Fail Instance 1 to Node 1. This will complete failover testing for Instance 1 and assuming no problems, it can be opened for other testing and/or production usage.	Any	6.ps1
22	Fail Instance 3 to Node 1.	Any	6.ps1
23	Fail Instance 3 to Node 2.	Any	6.ps1
24	Fail Instance 3 to Node 3. This will complete failover testing for Instance 3 and assuming no problems, it can be opened for other testing and/or production usage.	Any	6.ps1



# Conclusion for Scenario Two

---

Clearly this scenario is much easier and faster. Not only do you have more flexibility with the addition of another node, but the downtime and impact to applications and end users is minimized because this scenario is designed to minimize costly reboots and failovers. Initially, only one instance is impacted (in this case, Instance 3) with an outage. However, assuming you have the capacity on those two nodes that will own the instances (Node A and Node B), you not only have places to fail over, but you can complete the upgrade at a later date if your maintenance window is short or you just can't perform the failover test for other reasons. The overall risk is minimal, especially compared to the previous scenario where no matter what you do, you are shuffling things around and have multiple reboots.

# About the Author

---

Allan Hirt has been using SQL Server in various guises since 1992 and is currently a Microsoft Cluster MVP. For the past 10 years, he has traveled all over the world to train, mentor, and work with clients. He has also authored or contributed to publications about SQL Server for Microsoft, Penton Media (publishers of *SQL Server Magazine*), and others via white papers, webcasts, training courses, and books. He is a frequent speaker at conferences like TechEd and SQL PASS. His most recent book, "Pro SQL Server 2008 Failover Clustering," was published by Apress in July, 2009. He is currently working on an update for the next version of SQL Server, which is codenamed Denali. His blog and contact information can be found at <http://www.sqlha.com> and on Twitter via the handle @SQLHA.

© 2010 Quest Software, Inc.  
**ALL RIGHTS RESERVED.**

This document contains proprietary information protected by copyright. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording for any purpose without the written permission of Quest Software, Inc. (“Quest”).

The information in this document is provided in connection with Quest products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Quest products. EXCEPT AS SET FORTH IN QUEST'S TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT, QUEST ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL QUEST BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF QUEST HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Quest makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Quest does not make any commitment to update the information contained in this document.

If you have any questions regarding your potential use of this material, contact:

Quest Software World Headquarters  
LEGAL Dept  
5 Polaris Way  
Aliso Viejo, CA 92656  
**www.quest.com**  
email: **legal@quest.com**

Refer to our Web site for regional and international office information.

## **Trademarks**

Quest, Quest Software, the Quest Software logo, AccessManager, ActiveRoles, Aelita, Akonix, AppAssure, Benchmark Factory, Big Brother, BridgeAccess, BridgeAutoEscalate, BridgeSearch, BridgeTrak, BusinessInsight, ChangeAuditor, ChangeManager, Defender, DeployDirector, Desktop Authority, DirectoryAnalyzer, DirectoryTroubleshooter, DS Analyzer, DS Expert, Foglight, GPOAdmin, Help Desk Authority, Imceda, IntelliProfile, InTrust, Invirtus, iToken, IWatch, JClass, Jint, JProbe, LeccoTech, LiteSpeed, LiveReorg, LogADmin, MessageStats, Monosphere, MultSess, NBSpool, NetBase, NetControl, Npulse, NetPro, PassGo, PerformaSure, Point,Click,Done!, PowerGUI, Quest Central, Quest vToolkit, Quest vWorkSpace, ReportADmin, RestoreADmin, ScriptLogic, Security Lifecycle Map, SelfServiceADmin, SharePlex, Sitraka, SmartAlarm, Spotlight, SQL Navigator, SQL Watch, SQLLab, Stat, StealthCollect, Storage Horizon, Tag and Follow, Toad, T.O.A.D., Toad World, vAutomator, vControl, vConverter, vFoglight, vOptimizer, vRanger, Vintela, Virtual DBA, VizionCore, Vizioncore vAutomation Suite, Vizioncore vBackup, Vizioncore vEssentials, Vizioncore vMigrator, Vizioncore vReplicator, WebDefender, Webthority, Xaffire, and XRT are trademarks and registered trademarks of Quest Software, Inc in the United States of America and other countries. Other trademarks and registered trademarks used in this guide are property of their respective owners.

Updated—April, 2011

## About Quest Software, Inc.

Quest Software (Nasdaq: QSFT) simplifies and reduces the cost of managing IT for more than 100,000 customers worldwide. Our innovative solutions make solving the toughest IT management problems easier, enabling customers to save time and money across physical, virtual and cloud environments. For more information about Quest solutions for application management, database management, Windows management, virtualization management and IT management, go to [www.quest.com](http://www.quest.com).

## Contacting Quest Software

PHONE 800.306.9329 (United States and Canada)

If you are located outside North America, you can find your local office information on our Web site.

EMAIL [sales@quest.com](mailto:sales@quest.com)

MAIL Quest Software, Inc.  
World Headquarters  
5 Polaris Way  
Aliso Viejo, CA 92656  
USA

## Contacting Quest Support

Quest Support is available to customers who have a trial version of a Quest product or who have purchased a commercial version and have a valid maintenance contract.

Quest Support provides around-the-clock coverage with SupportLink, our Web self-service. Visit SupportLink at <https://support.quest.com>.

SupportLink gives users of Quest Software products the ability to:

- Search Quest's online Knowledgebase
- Download the latest releases, documentation and patches for Quest products
- Log support cases
- Manage existing support cases

View the Global Support Guide for a detailed explanation of support programs, online services, contact information and policies and procedures.



5 Polaris Way, Aliso Viejo, CA 92656 | PHONE 800.306.9329 | WEB [www.quest.com](http://www.quest.com) | EMAIL [sales@quest.com](mailto:sales@quest.com)

If you are located outside North America, you can find local office information on our Web site.

© 2011 Quest Software, Inc.  
ALL RIGHTS RESERVED.

Quest, Quest Software, the Quest Software logo [and product name] are registered trademarks of Quest Software, Inc. in the U.S.A. and/or other countries. All other trademarks and registered trademarks are property of their respective owners. WPD-Cluster-SQL-Server-US-KS-20110407