

**Patching SQL Server Failover Cluster Instances with Cluster-Aware Updating (CAU)**

SQL Server Technical Article

**Writer:** Dhawa Ngoche

**Technical Reviewer:** Gail Erickson, Shamik Gosh, Russell Green, Tim Wong, Min He

**Published:** January 2013

**Applies to:** SQL Server 2012 on Windows Server 2012

**Summary:** This paper describes the process of applying software updates on SQL Server 2012 failover cluster instances by using the Cluster-Aware Updating (CAU) feature that is available in Windows Server 2012. CAU’s update scheduling features and [Windows PowerShell Cluster-Aware Updating-specific cmdlets](http://technet.microsoft.com/en-us/library/hh847221.aspx) (http://technet.microsoft.com/en-us/library/hh847221.aspx) provide a very rich and flexible set of capabilities that can eliminate the lengthy manual processes that are usually required when applying updates to SQL Server failover cluster instances. Samples and demonstrations in this paper use SQL Server 2012 Service Pack 1 as the software update being applied as it is the first Microsoft SQL Server update that supports CAU.

For more information about CAU, see [Cluster-Aware Updating Overview](http://technet.microsoft.com/library/hh831694.aspx). The [Additional Resources](#AdditionalResources) section of this paper also provides links to the CAU documentation.

Copyright

This document is provided “as-is”. Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

© 2013 Microsoft. All rights reserved.

Contents

[1 Introduction 4](#_Toc345672683)

[1.1 How SQL Server 2012 Service Pack 1 Enables CAU Support 4](#_Toc345672684)

[1.2 Prerequisites 4](#_Toc345672685)

[2 CAU Plug-ins Overview 4](#_Toc345672686)

[2.1 Microsoft.HotfixPlugin 5](#_Toc345672687)

[2.2 Microsoft.WindowsUpdatePlugin 5](#_Toc345672688)

[3 Using Microsoft.HotfixPlugin 5](#_Toc345672689)

[3.1 Publish the update to a CAU directory structure 6](#_Toc345672690)

[3.1.1 Example 1: Node-specific update directory structure 6](#_Toc345672691)

[3.2 Create a Custom Hotfix Configuration File 7](#_Toc345672692)

[3.3 Execute the Updating Run using Microsoft.HotfixPlugin 11](#_Toc345672693)

[4 Using Microsoft.WindowsUpdatePlugin 15](#_Toc345672694)

[4.1 WSUS Server Setup 15](#_Toc345672695)

[4.2 WSUS Client Setup 15](#_Toc345672696)

[4.3 Executing the Updating Run using Microsoft.WindowsUpdatePlugin 17](#_Toc345672697)

[5 Plug-in Comparison Table 20](#_Toc345672698)

[6 Sample Output of the CAU Result Report 21](#_Toc345672699)

[7 How CAU Affects SQL Server AlwaysOn Availability Groups 24](#_Toc345672700)

[8 Troubleshooting 25](#_Toc345672701)

[9 Best Practices 26](#_Toc345672702)

[10 Additional Resources 27](#_Toc345672703)

[11 Conclusion 27](#_Toc345672704)

# Introduction

Starting with the update to Microsoft SQL Server 2012 Service Pack 1, customers with SQL Server 2012 failover cluster instances can greatly benefit from the fully automated update process provided by Cluster-Aware Updating (CAU).

CAU causes minimal down time for SQL Server 2012 client connectivity. While the primary node is being updated, the SQL Server clustered role (also called the resource group) goes through two failovers. The first failover happens before updating the primary node when CAU brings the original primary node into maintenance mode. During the update process on the original primary node, the SQL clustered role is temporarily brought online on a new primary node. The second failover happens once the update process is complete on the original primary node, and CAU fails back the SQL Server clustered role to the original primary node and brings it online.

## How SQL Server 2012 Service Pack 1 Enables CAU Support

To support CAU, the following changes were made in the SQL Server 2012 Service Pack 1 setup.

* Setup detects when CAU is deploying SQL Server 2012 Service Pack 1 on a node in maintenance mode.
* Supports running SQL Server 2012 Service Pack 1 setup under the LocalSystem account through CAU deployment by disabling detection of the remote machine state.

**Note:** These changes, which facilitate CAU support, are not available in any SQL Server update earlier than SQL Server 2012 Service Pack 1.

## Prerequisites

Having a good understanding of CAU technology is required to understand the information provided in this paper. We recommend that you read the following information before proceeding further.

* [Cluster-Aware Updating](http://technet.microsoft.com/en-us/library/hh831694.aspx) (http://technet.microsoft.com/library/hh831694.aspx)
* [Requirements and Best Practices for Cluster-Aware Updating](http://technet.microsoft.com/library/jj134234.aspx) (http://technet.microsoft.com/en-us/library/jj134234.aspx).

# CAU Plug-ins Overview

CAU provides two plug-ins: Microsoft.HotfixPlugin and Microsoft.WindowsUpdatePlugin. SQL Server 2012 Service Pack 1 and subsequent cumulative updates (CU) and hotfixes can be applied using either of these plug-ins, however, each plug-in has specific advantages and disadvantages that are described in this section.

To choose the appropriate plug-in, you must evaluate your requirements and cluster configuration environment. See the section [Plug-in Comparison Table](#_Plug-in_Comparison_Table) to understand the advantages and disadvantages of each plug-in with respect to updating SQL Server failover cluster instances. Regardless of which plug-in you use, the update process can be performed by using either the CAU user interface (UI) or the Windows PowerShell CAU cmdlets. For more information, see [How CAU Plug-ins Work](http://technet.microsoft.com/library/jj134213.aspx) (http://technet.microsoft.com/library/jj134213.aspx).

## Microsoft.HotfixPlugin

The typical user scenarios that are applicable to the Microsoft.HotfixPlugin include the following.

1. Updating **all** SQL Server 2012 cluster instances across **all** nodes using either Remote-Updating-Mode, or Self-Updating-Mode.
2. Updating a **specific** SQL Server 2012 cluster instance across **all** nodes using either Remote-Updating-Mode, or Self-Updating-Mode.
3. Updating a **specific** SQL Server 2012 cluster instance on **specific** nodes. This capability provides further granularity for control.

The advantage of Microsoft.HotfixPlugin is that it supports updating **specific** SQL Server 2012 failover cluster instances. This functionality provides SQL Server administrators with the ability to choose which instances to update and leave other instances intact. This functionality is not supported through Microsoft.WindowsUpdatePlugin.

The disadvantages of Microsoft.HotfixPlugin are:

* It does not support applying multiple updates that should be installed in sequence.
* It does not support a detection logic mechanism to determine update applicability. Hence, with every Updating Run, the updates are offered and executed even if the update is already installed.

## Microsoft.WindowsUpdatePlugin

The typical user scenarios that are applicable to the Microsoft.WindowUpdatePlugin include the following.

1. Updating all SQL Server 2012 RTM cluster instances with a single update such as Service Pack 1 by using either Remote-Updating-Mode, or Self-Updating-Mode.
2. Updating all SQL Server 2012 cluster instances with sequential updates such as Service Pack 1 and Service Pack 1 Cumulative Update (CU) by using either Remote-Updating-Mode, or Self-Updating-Mode.

The advantage of Microsoft.WindowsUpdatePlugin is that it supports applying updates sequentially.

Regardless of which plugin you choose, CAU provides several run options that you can customize as appropriate for your cluster, or class of clusters (for example, Business Critical Clusters). You can save those settings in a run profile to be used repeatedly to achieve a consistent updating process experience. For more information, see [Advanced Options and Updating Run Profiles for CAU](http://technet.microsoft.com/en-us/library/jj134224.aspx) (http://technet.microsoft.com/en-us/library/jj134224.aspx).

# Using Microsoft.HotfixPlugin

While the CAU documentation explains that this plugin can be used for non-Microsoft updates, it offers greater flexibility than Microsoft.WindowsUpdatePlugin in terms of updating SQL Server 2012 failover cluster instances because you can update specific SQL Server 2012 cluster instance(s), or apply an update on a selective set of cluster nodes. With Microsoft.WindowsUpdatePlugin, you don’t have this flexibility. However, as noted above, Microsoft.WindowsUpdatePlugin has specific advantages over Microsoft.HotfixPlugin. So, one must evaluate the requirements and cluster ecosystem in order to choose which plugin to use.

This plug-in is suitable for applying a single hotfix update only once with the understanding that the update will be rerun if you rerun the CAU Updating Run.

If an update is rerun, even though the SQL Server instance already has the update, setup will exit (without reinstalling) with a specific exit code, which must be handled in the CAU hotfix configuration file so that CAU doesn’t treat the run as a failure. Similarly, if the update is run on a node that doesn’t have an applicable SQL Server 2012 instance, CAU will treat it as a failure unless the specific exit code is captured in the CAU hotfix configuration file. Section 3.2 below shows the actual exit codes returned by SQL Server 2012 update for the exit conditions **NotApplicable** and **AlreadyInstalled**.

Using the CAU directory structure, if you put an update in a folder named after a node, the update will be run on that node only. Similarly, if you have an update under a set of folders named after the nodes, the update will be run sequentially on all the nodes having a corresponding update folder.

To use the Microsoft.HotfixPlugin, the following steps are required.

1. Publish the update to a CAU directory structure
2. Create a custom hotfix configuration file
3. Execute the Updating Run by using Microsoft.HotfixPlugin

## Publish the update to a CAU directory structure

The steps in this section explain how to create the CAU directory structure that is required to apply updates to specific SQL Server 2012 cluster instances.

1. Review [How CAU Plug-ins Works](http://technet.microsoft.com/en-us/library/jj134213.aspx) (http://technet.microsoft.com/en-us/library/jj134213.aspx), which explains how the CAU hotfix directory structure can be used for different needs.
2. Create a network share for the directory structure (for example, \\<networkshare>\hotfixroot). This can be a standard network share, or a Server Message Block (SMB)-enabled file share.

When a Windows Server 2012 cluster hosts several SQL Server failover cluster instances, it is important that there is little or no impact to any non-targeted SQL Server instances when you apply an update. The following example shows how to create a CAU directory structure that provides the highest level of control and precision.

### Example 1: Node-specific update directory structure

Assume that you have eight nodes in your Windows Server 2012 cluster and four SQL Server 2012 failover cluster instances. Each failover cluster instance uses two nodes and each node is part of no more than one SQL Server instance. The following table shows this node distribution.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Node1** | **Node2** | **Node3** | **Node4** | **Node5** | **Node6** | **Node7** | **Node8** |
| SQLInst1 | | SQLInst2 | | SQLInst3 | | SQLInst4 | |

**Table 1:** Node distribution

Assume that the update requirement specifies that only SQL Server instances **SQLInst1** and **SQLInst2** should be updated with SQL Server 2012 Service Pack 1 and but not instances **SQLInst3** and **SQLinst4**.

Given the above scenario, the following steps are required.

1. To prevent an accidental update on all the nodes, remove or rename the "CAUHotfix\_All" folder under \\<networkshare>\hotfixroot\if it exists.
2. Create a folder named after each of the nodes directly under the "hotfixroot" folder.
3. Create a subfolder inside the node name folder (for example, "SQLServer2012SP1") for each node.
4. Copy the update package into the subfolders. In this example, copy the SQL Server 2012 Service Pack 1 package so that directory structure appears as follows.

\\<networkshare>\hotfixroot

\Node1

           \SQLServer2012Sp1\<SQLServer2012SP1Package>.exe

\Node2

           \SQLServer2012Sp1\<SQLServer2012SP1Package>.exe

\Node3

           \SQLServer2012Sp1\<SQLServer2012SP1Package>.exe

\Node4

               \SQLServer2012Sp1\<SQLServer2012SP1Package>.exe

## Create a Custom Hotfix Configuration File

The instructions in this section are based on the scenario and requirements defined in **Example 1** above.

1. Read the section 'Customize the Hotfix Configuration File' in the topic [How CAU Plug-ins Work](http://technet.microsoft.com/library/jj134213.aspx) to learn more about the DefaultHotfixConfig.xml file. This file must exist under the "hotfixroot" folder (for example, \\<networkshare>\hotfixroot\DefaultHotfixConfig.xml) because it contains the default rules that are used to handle common update extensions (for example, .msi, .msu, and .msp).
2. Copy the configuration file from any of the cluster nodes to the "hotfixroot" folder on the share. You can find it under C:\Windows\System32\WindowsPowerShell\v1.0\Modules\ClusterAwareUpdating on any of the cluster nodes, or on the orchestrator machine.
3. To update only **SQLInst1**, and **SQLInst2**, make two copies of DefaultHotfixConfig.xml file and name the files **SQLInst1Config.xml** and **SQLInst2Config.xml**. These custom hotfix configuration files must also exist under the "hotfixroot" folder. See the CAU documentation regarding security considerations and the importance of securing configuration files so that they cannot be tampered with by malicious users. Custom configuration files contain information such as your update folder name (for example, **SQLServer2012SP1**) as well as command-line parameters and expected exit codes.
4. Modify the **SqlInst1Config.xml** and **SQLInst2Config.xml** files by inserting the **FolderRules** element under the **DefaultRules** element. The following sample code uses the command-line to install SQL Server 2012 Service Pack 1 on a SQL Server instance named **SQLInst1**.

<root>

<DefaultRules>

…

</DefaultRules>

<FolderRules>

<Folder name="**SQLServer2012SP1**">

<Template path="$update$" parameters="**/ACTION=PATCH /INSTANCENAME=SQLInst1 /QUIET /IAcceptSQLServerLicenseTerms**"/>

<ExitConditions>

<Success>

<ExitCode code="**0**"/>

</Success>

<Success\_RebootRequired>

<ExitCode code="**3010**"/>

</Success\_RebootRequired>

<NotApplicable>

<ExitCode code="**-2068578302**"/> <!-- ERROR\_PATCH\_TARGET\_NOT\_FOUND -->

</NotApplicable>

<AlreadyInstalled>

<ExitCode code="**-2068643838**"/> <!-- ERROR\_PATCH\_ALREADY\_APPLIED -->

</ AlreadyInstalled >

</ExitConditions>

</Folder>

</FolderRules>

</root>

**Additional information about the previous highlighted code**

* You can name the update folder any name, but the <folder name> attribute value ("SQLServer2012SP1" in this example) must match the folder name under <nodename> (for example, "Node1\**SQLServer2012SP1**") on the hotfix share.
* The name of the SQL Server instance in the command-line must be changed to the instance you want to update. We recommend dedicating one configuration file per SQL Server instance to provide visual clarity.
* Exit code "**0**" is returned by the SQL Server update operation when the update succeeds and no reboot is required.
* Exit code "**3010**" is the standard exit code returned by the update process when a reboot is required after a successful update installation.
* **NotApplicable** exit code "**-2068578302**" is returned by the SQL Server update process if the target SQL Server instance is not found on a given node. Therefore, handling this exit code in the configuration file will treat the result as a success when the node might not have the target SQL Server instance. Do NOT handle this exit code if your directory structure is using node name folders and you need to detect invalid updating runs. However, if you are using the "**CAUHotfix\_All**" folder with a cluster environment in which the target SQL Server instance or instances might exist on a subset of the nodes and you don’t care if the update is run on nodes that don’t have the target SQL Server instance.
* **Alreadyinstalled** exit code "**-2068643838**" is returned by SQL Server updates if the update is already installed and the update is being reinstalled. Because Microsoft.HotfixPlugin does not support a detection logic mechanism to determine if an update should be offered, each Updating Run process using Microsoft.HotfixPlugin will result in the reinstallation of the update. So, handling this exit code in the configuration file treats all reinstalls as success.

Any exit code that is not handled is treated as "Failed". The following **ExitcodeConditions** are supported by CAU:

* Success
* Success\_RebootRequired
* NotApplicable
* AlreadyInstalled.

You cannot arbitrarily define exit code conditions.

**Scenario Examples**

The following example scenarios show some configuration variations and update command-line uses. It assumes that the CAU directory structure uses a folder named after each node and is not "CAUHotfix\_All".

**Scenario 1:**

On an 8-node cluster, the SQL Server instance **Inst1** exists on the odd number nodes. Even number nodes (not shown in the table) have other SQL Server instances.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nodes | Node1 | Node3 | Node5 | Node7 |
| SQL Instance | Inst1 | Inst1 | Inst1 | Inst1 |

**Table 2**: Instances on odd-numbered nodes

Requirement: All nodes having the **Inst1** instance must be updated to SQL Server 2012 Service Pack 1.

Solution:

1. Create a CAU directory structure with the SQL Server 2012 Service Pack 1 package.
2. Create a custom hotfix configuration file that contains the following: "/action=patch **/allinstances** /quiet /IAacceptSQLServerLicenceTerms", or "/action=patch **/InstanceName=Inst1** /quiet /IAacceptSQLServerLicenceTerms".

Because the non-target SQL Server instances are on other nodes, they will remain intact.

**Scenario 2**:

On an 8-node cluster, odd number nodes have a mix of SQL Server failover cluster instances. Even number nodes (not shown in the table) have other SQL Server instances.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nodes | Node1 | Node3 | Node5 | Node7 |
| SQL Instance | Inst1, Inst2 | Inst1 | Inst1 | Inst1, Inst2 |

**Table 3**: Mixed instances on odd-numbered nodes

Requirement:Both instances **Inst1** and instance **Inst2** must be patched with Service Pack 1.

Solution:

1. Create a CAU directory structure with the SQL Server 2012 Service Pack 1 package.
2. You have two choices when creating the custom hotfix configuration file:
   1. Create a single custom configuration file with "/action=patch **/allinstances** /quiet /IAacceptSQLServerLicenceTerms".
   2. Create two configuration files, one that has "/action=patch **/InstanceName=Inst1** /quiet /IAacceptSQLServerLicenceTerms", and another file that has "/action=patch **/InstanceName=Inst2** /quiet /IAacceptSQLServerLicenceTerms".

**Note**: Option b requires that you run CAU two times using the respective configuration file for each instance. We recommend that you use option a because it will be faster. Because the non-target SQL Server instances are on other nodes, they will remain intact.

**Scenario 3:**

On an 8-node cluster, odd number nodes have a mix of SQL Server failover cluster instances. Even number nodes (not shown in the table) have other SQL Server instances. This scenario demonstrates the effect of the CAU process on non-target instances when a node is shared.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nodes | Node1 | Node3 | Node5 | Node7 |
| SQL Instance | Inst1, Inst2, Inst3 | Inst1, Inst2 | Inst1, Inst2 | Inst1, Inst2 |

**Table 4**: Mixed instances on odd-numbered nodes with a non-targeted instance

Requirement: Instance **Inst1**, and **Inst2** must be updated with SQL Server 2012 Service Pack 1, but instance **Inst3** must remain intact.

Solution:

1. Create a CAU directory structure with the SQL Server 2012 Service Pack 1 package.
2. Create two custom hotfix configuration files, one having "/action=patch **/InstanceName=Inst1** /quiet /IAacceptSQLServerLicenceTerms", and the other one having "/action=patch **/InstanceName=Inst2** /quiet /IAacceptSQLServerLicenceTerms". This means that you need to run CAU two times using the respective custom hotfix configuration file for each instance.

**Caveat**:While only instance **Inst1** and **Inst2** will be updated and not **Inst3**, instance **Inst3** will experience downtime when CAU brings Node1 into maintenance mode to apply the update to **Inst1** and **Inst2**.

## Execute the Updating Run using Microsoft.HotfixPlugin

CAU Updating Run can be executed by using either Windows PowerShell or the CAU UI tool with Remote-Updating-Mode or Self-Updating-Mode. The following table shows examples of each method.

|  |  |
| --- | --- |
| **Remote-Updating Mode (Microsoft.HotfixPlugin)** | |
| **General Steps for Remote-Updating Mode**   1. Set up a machine (referred to as the orchestrator) that is not part of the cluster. This machine must be either a Windows 8 client or a Windows Server 2012 server. It cannot be one of the cluster nodes or a machine with an earlier operating system version. 2. Log on to the orchestrator and launch a Windows PowerShell console window in elevated mode. 3. Run the Windows PowerShell cmdlet Add-WindowsFeature RSAT-Clustering-Mgmt. This will install the Failover Clustering Tools that exists under Remote Server Administration. Failover Clustering tools contain the Windows Failover Cluster Manager and Cluster-Aware Updating. | |
| **Invoke Remote Updating by Using CAU UI Tool** | **Invoke Remote Updating by Using CAU Cmdlets** |
| 1. From **Server Manager** | **Tools**, launch **Cluster-Aware Updating**. 2. Connect to the target cluster to be updated.   **Note**: An authentication dialog will pop up if the login for the orchestrator machine is different than the administrative account on the cluster nodes.   1. Use Microsoft.HotfixPlugin with appropriate settings (for example, Plugin-arguments, see the Windows PowerShell example in the right panel) and execute the **Updating Run**.   **Note**: The default plug-in is Microsoft.WindowsUpdatePlugin.   1. Update progress should begin. Let it continue until it completes. 2. Once the update process is complete, review the CAU report on the Updating Run by clicking **Generate Report on the Past Updating Runs**. | 1. Run the following sample command, which uses the parameters $cluster=$cluster.Name and $cred=get-credential.   You must supply a user name and password. The credential parameter is required if the login for the orchestrator machine is different than the administrative account on the cluster nodes.  Invoke-CauRun -ClusterName $cluster.Name -Force -CauPluginArguments @{ 'HotfixRootFolderPath' = '\\<networkshare>\HotfixRoot'; 'DisableAclChecks' = 'True'; 'RequireSmbEncryption' = 'False'; 'HotfixConfigFileName' = 'SQLInst1Config.xml' } -CauPluginName Microsoft.HotfixPlugin -MaxRetriesPerNode 1 –cred $cred  Update progress should begin.  The following cmdlet example can be used to monitor the status of the current updating runs.  get-caurun -ClusterName $cluster.Name –WaitForCompletion  The WaitForCompletion parameter returns after the Updating Run has finished, and writes the updated CAU run objects to the output stream as progress is recorded by the current CAU Update Coordinator.  The following cmdlet example retrieves the Updating Run report and can be used to troubleshoot failures.  $report = Get-CauReport -ClusterName $cluster.Name -Detailed –Last –cred $cred  For more information, see the [Troubleshooting](#Troubleshooting) section of this paper for sample interaction with the result report. |

**Table 5:** Hotfix Plugin using Remote-Updating Mode example

|  |  |
| --- | --- |
| **Self-Updating Mode (HotfixPlugin)** | |
| **General Steps for Self-Updating Mode:**   1. Log on to each of the cluster nodes. 2. Launch a Windows PowerShell console window in admin mode. 3. Run the Windows PowerShell cmdlet Add-WindowsFeature RSAT-Clustering-Mgmt on each of the cluster nodes. This will install the Failover Clustering Tools that are under the Remote Server Administration. Failover Clustering tools contain the **Windows Failover Cluster Manager** and **Cluster-Aware Updating**. 4. Run the following command from **only one** of the nodes to enable Self-Updating Mode on the CAU role on the cluster. Do **not** run this on all of the nodes.   Add-CauClusterRole $cluster.Name -EnableFirewallRules | |
| **Invoke Self-Updating by Using CAU UI Tool** | **Invoke Self-Updating by Using CAU Cmdlets** |
| 1. Log on to one of the cluster nodes. 2. From **Server Manage**r | **Tools**, launch **Cluster-Aware Updating**. 3. Connect to the cluster. You can use "." to specify the current cluster. 4. Use Microsoft.HotfixPlugin with appropriate settings (for example, Plug-in-arguments, see the Windows PowerShell example in the right panel) and execute the Updating Run.   **Note:** Default plug-in is Microsoft.WindowsUpdatePlugin.   1. Update progress should begin. Let it continue until it completes. 2. Once the update process is complete, review the CAU report on the Updating Run by clicking **Generate Report on the Past Updating Runs**. | 1. Log on to one of the cluster nodes and launch a Windows PowerShell console window. 2. Run the following commands.   **Note**: You can set self-updates to run on a weekly or monthly schedule. The following example show how to override the default plug-in with the HotifixPlugin and then starts the update immediately using by specifying the -UpdateNow switch. This switch cannot be run with –caupluginname or -caupluginargments. Because of this, you must first run the cmdlet to configure which plug-in to use and then run it again to run CAU immediately using -UpdateNow switch.  **First Run:**  Set-CauClusterRole -clustername $cluster.Name -caupluginname Microsoft.HotfixPlugin -caupluginarguments @{ 'HotfixRootFolderPath' = '\\ networkshare>\HotfixRoot'; 'DisableAclChecks' = 'True'; 'RequireSmbEncryption' = 'False'; 'HotfixConfigFileName' = 'SQLInst1Config.xml' }  Note: The HotfixConfigFileName parameter is used to specify which custom configuration file to use.  **Second Run:**  Set-CauClusterRole -ClusterName $cluster.Name -UpdateNow -Force  Update progress should begin.  The following cmdlet example can be used to monitor the status of the current updating runs.  get-caurun–WaitForCompletion  The following cmdlet example retrieves the Updating Run report and can be used to troubleshoot failures.  $report = Get-CauReport -ClusterName $cluster.Name -Detailed –Last  For more information, see the [Troubleshooting](#Troubleshooting) section of this paper for sample interaction with the result report. |

**Table 6:** Hotfix Plugin using Self-Updating Mode example

**Note**: Whether your update is under the "CAUHotfix\_All" folder or under a specific folder named after a node, the configuration file must contain the correct folder name for each update. If the folder name exists only on the share and is not contained in the configuration file, that folder is ignored.

# Using Microsoft.WindowsUpdatePlugin

Read the CAU [documentation](http://technet.microsoft.com/en-us/library/jj134213.aspx#BKMK_WUP) (http://technet.microsoft.com/en-us/library/jj134213.aspx#BKMK\_WUP) for this plug-in. Also, if you are not familiar with either Windows Update or Windows Server Update Services (WSUS) server setup, refer to the online resources for technical details. It is beyond the scope of this paper to cover these technologies in detail. However, the following sections provide the known requirements for WSUS server setup and step-by-step instructions to configure the WSUS client.

## WSUS Server Setup

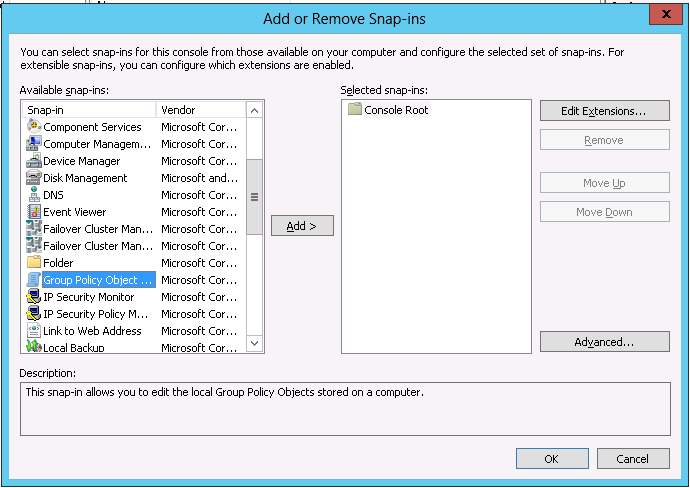
Install <http://support.microsoft.com/kb/2734608> on a WSUS server. This update allows servers that are running Windows Server Update Services (WSUS) 3.0 Service Pack 2 to provide updates to computers that are running Windows 8 or Windows Server 2012.

## WSUS Client Setup

In order to receive updates from your internal WSUS server, Windows 8 and Windows Server 2012 WSUS clients must point to your WSUS server in the same way as Windows 7 or earlier operating system clients. Refer to the section "To point the client computers to the WSUS server" in the topic [Step 5: Configure client updates](http://technet.microsoft.com/library/dd939830(v=ws.10).aspx) (http://technet.microsoft.com/library/dd939830(v=ws.10).aspx).

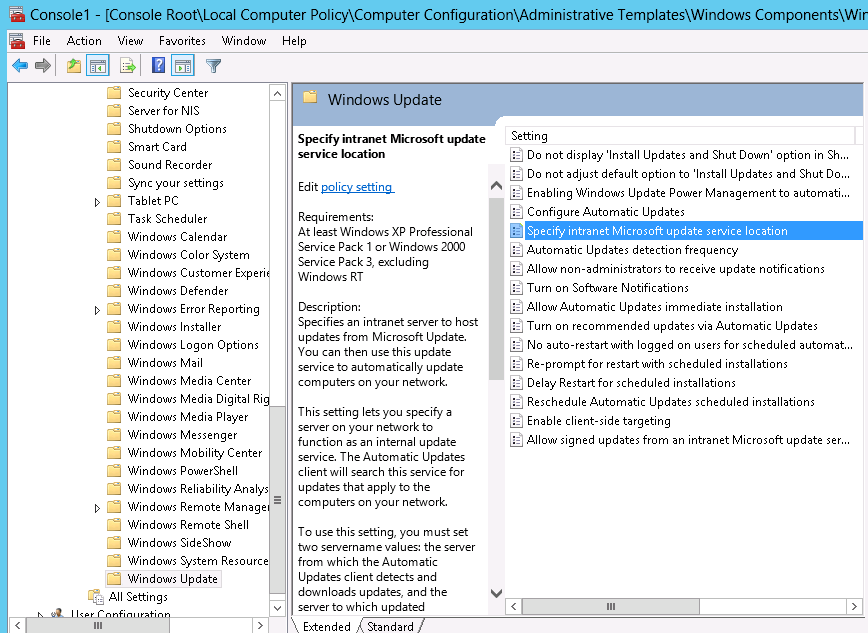
The following instructions can be used to configure the WSUS client.

1. From a console window (command prompt window), launch **mmc** or address field from your cluster node.
2. From the **File** menu, choose **Add/Remove Snap-in**.
3. From **Add or Remove Snap-ins**, choose **Group Policy Object** and click **Add**.



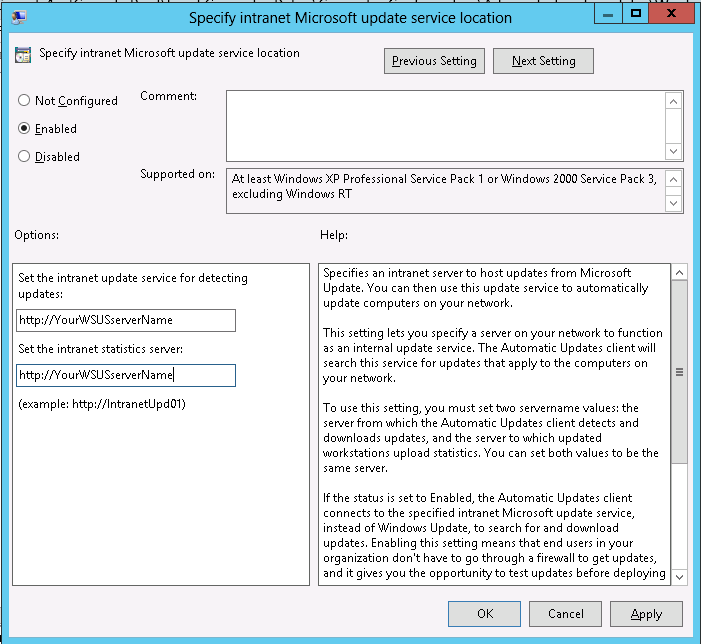
**Figure 1**: Add or Remove Snap-ins Dialog

1. Verify **Local Computer** is set in the **Group Policy Object** and click **Finish**.
2. Click **OK** to exit the **Add or Remove Snap-ins** dialog.
3. From **Console Root**, expand **Local Computer Policy** | **Administrative Templates** | **Windows Components** | **Windows Update.**
4. Under **Windows Update**, double-click **Specify intranet Microsoft Update service location**.



**Figure 2**: Windows Update service location

1. Select **Enabled** and then enter the URL for your WSUS server (for example, http://YourWsusServerName) in both the **Set the intranet update service for detecting updates** and **Set the intranet statistics server** fields.



**Figure 3**: Specify intranet Microsoft update service location dialog

1. Click **OK** and close the **mmc** console.
2. It may take up to 20 minutes for the group policy to take effect. To force the update to take immediately, run **gpupdate /force**.
3. To see if updates are being offered, you can launch **Windows Update**, and click **Check for updates**.

After you verify that SQL Server 2012 Service Pack 1 is being offered correctly on one of the cluster nodes, configure your orchestrator machine to execute the Updating Run process remotely as shown in the following section.

## Executing the Updating Run using Microsoft.WindowsUpdatePlugin

CAU Updating Run can be executed by using either Windows PowerShell cmdlets or the CAU UI tool with either Remote-Updating Mode or Self-Updating Mode. The following table shows examples of each method.

|  |  |
| --- | --- |
| **Remote-Updating Mode (Microsoft.WindowsUpdatePlugin)** | |
| **General Steps for Remote-Updating Mode:**   1. Set up a machine (referred to as the orchestrator) that is not part of the cluster. This machine can be a Windows 8 client machine. It cannot be one of the cluster nodes. 2. Log on to the orchestrator and launch a Windows PowerShell console window in elevated mode. 3. Run the Windows PowerShell cmdlet Add-WindowsFeature RSAT-Clustering-Mgmt. This will install the **Failover Clustering Tools** that exist under the **Remote Server Administration**. Failover Clustering Tools contain the **Windows Failover Cluster Manager** and **Cluster-Aware Updating**. | |
| **Invoke Remote Updating Using CAU UI Tool** | **Invoke Remote Updating Using CAU Cmdlets** |
| 1. From **Server Manager** | **Tools**, launch **Cluster-Aware Updating**. 2. Connect to the target cluster to be updated.   **Note**: An authentication dialog will pop up if the login for the orchestrator machine is different than the administrative account on the cluster node.   1. Use Microsoft.WindowsUpdatePlugin with the appropriate settings and execute the Updating Run. 2. Update progress should begin. Let it continue until completion. 3. Once the update process is complete, review the CAU report on the Updating Run by clicking **Generate Report on the Past Updating Runs**. | 1. Launch a Windows PowerShell console window in elevated mode. 2. Run following command, which uses the parameters $cluster=$cluster.Name and $cred=get-credential.   **Note:** You must supply a user name and password. The credential parameter is required if the login for the orchestrator machine is different than the administrative account on the cluster nodes.  Invoke-CauRun -ClusterName $cluster.Name -Force -CauPluginName Microsoft.WindowsUpdatePlugin -MaxRetriesPerNode 1 -CauPluginArguments @{ 'IncludeRecommendedUpdates' = 'False' } -EnableFirewallRules;  –cred $cred  Update progress should begin.  To view the progress of the update process, run the following command immediately after running the previous code. Alternatively, you can also use the CAU UI tool to view progress.  get-caurun –WaitForCompletion –cred $cred  Once the Updating Run completes, you can use the following commands to work with Updating Run report.  $report = Get-CauReport -ClusterName $cluster.Name -Detailed –Last –cred $cred  For examples on analyzing the report, see the section [Sample Output of CAU Result Report](#_Sample_Output_of). |

**Table 7:** WindowsUpdatePlugin using Remote-Updating Mode example

|  |  |
| --- | --- |
| **Self-Updating Mode (Microsoft.WindowsUpdatePlugin)** | |
| **General Steps for Self-Updating-Mode**   1. Log on to each of the cluster nodes. 2. Launch a Windows PowerShell console window in admin mode. 3. Run the Windows PowerShell cmdlet Add-WindowsFeature RSAT-Clustering-Mgmt on each of the cluster nodes. This will install the Failover Clustering Tools that exists under the Remote Server Administration. Failover Clustering Tools contain the Windows Failover Cluster Manager and Cluster-Aware Updating. 4. Run the following command from **only one** of the nodes to enable Self-Updating Mode on the CAU role on the cluster. Do **not** run it on all of the nodes.   Add-CauClusterRole $cluster.Name -EnableFirewallRules | |
| **Invoke Self-Updating Using CAU UI Tool** | **Invoke Self-Updating Using CAU Cmdlets** |
| 1. Log on to one of the cluster nodes. 2. From **Server Manager** | **Tools**, launch **Cluster-Aware Updating**. 3. Connect to the cluster. You can use " . " to specify the current cluster. 4. Use Microsoft.WindowsUpdatePlugin with the appropriate settings and execute the Updating Run. 5. Update progress should begin. Let it continue until it completes. 6. Once update process is complete, review the CAU report on the Updating Run by clicking **Generate Report on the PastUpdating Runs**. | 1. Log on to one of the cluster nodes and launch a Windows PowerShell console window. 2. Run the following commands.   **Note:** You can set self-updates to run on a weekly or monthly schedule. The example below shows how to run an on-demand update with two command-line executions. The first run sets Microsoft.WindowsUpdatePlugin as the current plug-in and the second command launches the CAU update immediately by using the -UpdateNow switch. This switch cannot be run with -caupluginname, -caupluginargments.  **First Run:**  Set-CauClusterRole -ClusterName $cluster.Name -Force -CauPluginName Microsoft.WindowsUpdatePlugin -MaxRetriesPerNode 1 -CauPluginArguments @{ 'IncludeRecommendedUpdates' = 'False' }  **Second Run:**  Set-CauClusterRole -ClusterName $cluster.Name -UpdateNow -Force  Run the following command immediately after running the previous on-demand update to view update progress. Alternatively, you can also view the update progress by using the CAU UI tool.  get-caurun –WaitForCompletion  Once the Updating Run completes, you can use the following commands to work with Updating Run report.  $report = Get-CauReport -ClusterName $cluster.Name -Detailed -Last  For examples on analyzing the report, see the section [Sample Output of CAU Result Report](#_Sample_Output_of). |

**Table 8:** WindowsUpdatePlugin using Self-Updating Mode example

# Plug-in Comparison Table

The following table compares the CAU plug-ins as they affect SQL Server failover clusters.

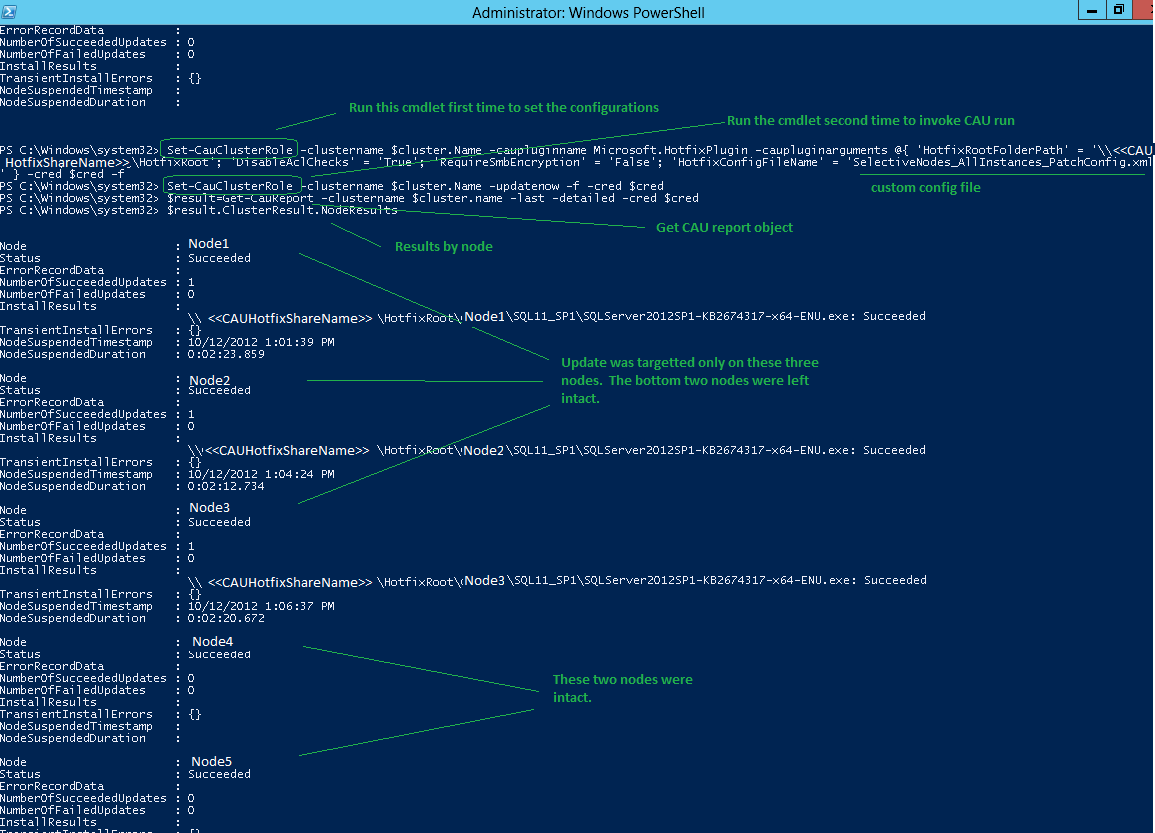
|  |  |  |
| --- | --- | --- |
| **Scenario** | **Microsoft.WindowsUpdatePlugin** | **Microsoft.HotfixPlugin** |
| Update **all** SQL Server 2012 cluster instances on all cluster nodes. | Supported | Supported |
| Update **all** SQL Server 2012 cluster instances on all cluster nodes with sequential updates (for example, SQL Server 2012 Service Pack 1 + Service Pack 1 Cumulative Update) | Supported | Not Supported  Sequential update not supported. |
| Update **specific** SQL Server 2012 cluster instance on **all** cluster nodes. | Not Supported  No way to specify a SQL instance name | Supported  Requires configuring a custom configuration file by specifying the SQL Server instance names to update. |
| Update **specific** SQL Server 2012 cluster instance on **specific** cluster node(s) | Not Supported  No way to choose selective nodes. | Supported  Requires configuring custom hotfix configuration file as well as CAU directory structure changes by replacing "CAUHotfix\_All" directory with node specific directory names. |

**Table 9:** Plug-in comparison

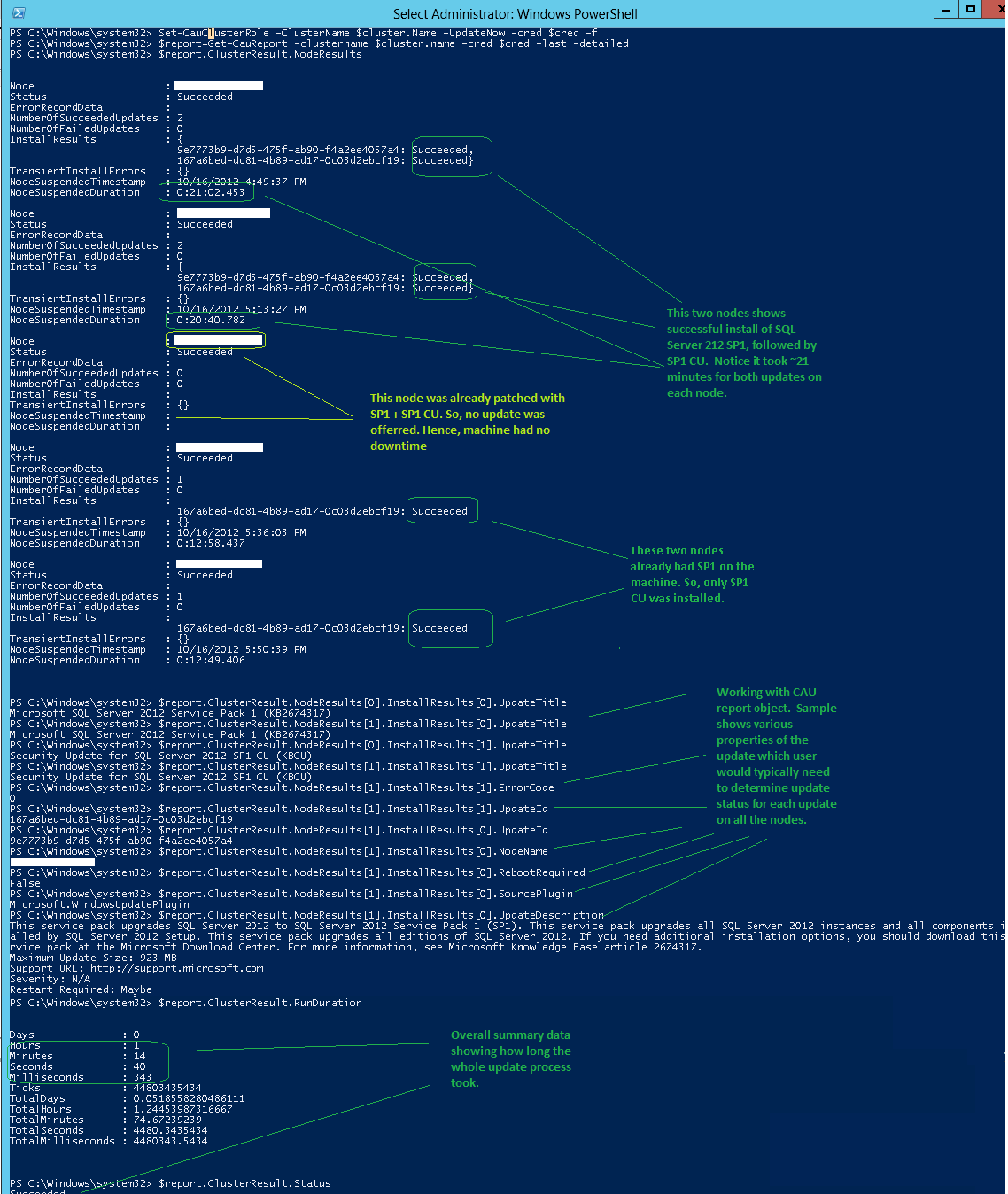
# Sample Output of the CAU Result Report

This sample output shows a successful installation of SQL Server 2012 Service Pack through on-demand Updating Run using the Self-Updating Mode. Updating Run uses a custom configuration file in combination with a hotfix directory structure named after the node to be updated. The annotation in Figure 4 shows that Node4 and Node5 remain intact as indicated by values for the properties NodeSuspendedTimestamp, and NodeSuspendedDuration.

**Note**: CAU status "Succeeded" does not necessarily mean the update was installed successfully. This status can also mean CAU ran successfully on a node regardless of whether an update was installed or not.



**Figure 4**: CAU Result Report output

Figure 5 that follows shows the result of running Microsoft.WindowsUpdatePlugin and also shows samples of results of the Updating Run report object by checking various properties. This figure demonstrates the ability to apply sequential updates and also shows how nodes that already have the required updates are left intact. **Note**: Node names in this sample are masked for security reasons.

**Figure 5**: CAU Result Report output for sequential updates

# How CAU Affects SQL Server AlwaysOn Availability Groups

**Disclaimer:** Microsoft does not support the use of CAU to update Windows Server 2012 clusters with SQL Server AlwaysOn availability groups. CAU is currently not aware of AlwaysOn availability groups. When CAU brings a node into maintenance mode to update the node, AlwaysOn availability groups are affected by the following known issues. Potential mitigations for these issues are provided to be transparent about our findings, and are meant for your **non-production test environment only**. These mitigations are not guaranteed to solve all issues.

1. Updating Run encounters node drain failure leaving the node unpatched if the primary replica is a standalone SQL Server instance. This happens when an AlwaysOn availability group has fewer than two replicas set to Automatic Failover mode. Note that without at least two replicas set to Automatic Failover mode, you cannot failover an AlwaysOn availability group even by using the Failover Cluster Manager tool.

**Possible Mitigation**: Set the primary replica and at least one other replica to Automatic Failover and run CAU again.

If the AlwaysOn availability group primary replica uses a SQL Server failover clustered instance using two or more nodes, then, although the AlwaysOn availability group will be temporarily forced to go offline when the SQL Server failover cluster instance fails over, it will come online as soon as the underlying SQL Server failover clustered instance comes online on another node. Furthermore, when the update is complete (for example, SQL Server 2012 Service Pack 1 takes about 15 minutes to install) on the original primary node, CAU will fail back the SQL Server failover clustered instance to the original primary node.

1. The AlwaysOn availability group randomly fails to come online because the target replica is not ready for failover. This is caused by a timing issue and is less likely to occur if you have three replicas with Automatic Failover mode. This happens more commonly in scenarios where the AlwaysOn availability group is on two replicas. For example, Node B was just updated and CAU proceeds to update node A (the primary replica), but the roles owned by node A are in the process of failing over to node B. This issue affects only AlwaysOn availability group replicas using stand-alone SQL Server instances.

**Possible Mitigation**: Manually bring the AlwaysOn availability group online on node B. If you don't, the AlwaysOn availability group will be down until the update on node A is complete (roughly 15 minutes for SQL Server 2012 Service Pack 1). At which point CAU will automatically fail back the AlwaysOn availability group to node A and it should come online.

The following examples further illustrate AlwaysOn availability group configurations and affecting issues:

1. Two replicas that are both on stand-alone SQL Server instances. Only one is in automatic failover.

|  |  |  |
| --- | --- | --- |
| **Replica** | **1 (AlwaysOn Availability Group Primary Replica)** | **2** |
| Failover Mode | Automatic | Manual |
| Cluster Node | A | B |
| SQL Server Instance | Standalone | Standalone |

**Table 10:** AlwaysOn Availability Group example 1

**CAU Result:** Replica 2 is updated first and when CAU tries to bring replica 1 into maintenance mode, it reports "Node drain failed on node <nodename>". This happens because Replica 2 is not set for automatic failover.

**Resulting behavior**: Node A is NOT patched, and the Availability Group goes into offline mode until Node A is manually brought back online.

**Possible Mitigation**: Set node B to Automatic Failover mode and run CAU again.

1. Two replicas are in automatic failover on the same subnet the third replica is in manual failover.

|  |  |  |  |
| --- | --- | --- | --- |
| **Replica** | **1 (AlwaysOn Availability Group Primary Replica)** | **2** | **3** |
| Failover Mode | Automatic | Automatic | Manual |
| Cluster Node | A | B | C |
| SQL Server Instance | Standalone | Standalone | Standalone |

**Table 11:** AlwaysOn Availability Group example 2

**CAU Result**: The Updating Run should succeed without hitting node drain failure. Caveat: Because there are only two nodes set to Automatic Failover mode, and if node B was updated just prior to bringing node A into maintenance mode, you might encounter a failure caused by a failover happening before the automatic failover target replica is ready.

**Possible Mitigation**: Use the CAU NodeOrder option to specify the node order B, C, and A. It should give node B ample time to be ready as a failover target. Review the CAU documentation for details on how to use the NodeOrder option.

1. Availability Group primary and the SQL failover primary are on the same node.

|  |  |  |  |
| --- | --- | --- | --- |
| **Replica** | **1 (AlwaysOn Availability Group Primary Replica)** | **2** | 3 |
| Failover Mode | Manual | Manual | Manual |
| Cluster Node | A (SQL primary), B | C (SQL primary), D | E |
| SQL Server Instance | Clustered | Clustered | Standalone |

**Table 12:** AlwaysOn Availability Group example 3

**CAU Result**: When CAU brings node A into maintenance mode the SQL Server failover instance comes online on node B. In this case, although the AlwaysOn availability resource group is **forced to go offline**, it is able to come online because node B has a SQL Server failover instance running. After node A is successfully updated, CAU fails back SQL instance to node A. In addition, the AlwaysOn availability resource group comes online again on node A.

# Troubleshooting

Troubleshooting can be complex because a failure can occur for various external reasons and not necessarily because of a SQL Server 2012 Service Pack 1 update. For example, failure could happen due to hotfix share configuration issues, machine readiness issues, Windows Update issues, WSUS client configuration issues, and update install issues. Refer to [Requirements and Best Practices for Cluster-Aware Updating](http://technet.microsoft.com/en-us/library/jj134234) for more information.

However, if CAU indicates a SQL update failure on a given node, login into the node and check the SQL Server setup logs under %programfiles%\Microsoft SQL Server\**110**\Setup Bootstrap\Log. Consult the following resources to understand how to locate and read SQL Server log files.

* [View and Read SQL Server Setup Log Files](http://technet.microsoft.com/en-us/library/ms143702(v=sql.110).aspx) (<http://technet.microsoft.com/en-us/library/ms143702(v=sql.110).aspx)>
* [How to identity SQL Server 2008 setup issues in the setup log files](http://support.microsoft.com/kb/955396) (<http://support.microsoft.com/kb/955396>). Although this resource is written against SQL Server 2008 setup, it is still valid for SQL Server 2012 setup logs. One difference is to look under folder path “**110**” (e.g. %programfiles%\Microsoft SQL Server\**110**\Setup Bootstrap\Log) for SQL Server 2012 setup.

To investigate CAU-related failures (in other words, non-SQL Server update failures), refer to [Troubleshooting Cluster-Aware Updating (CAU) in Windows Server 2012 clusters](http://social.technet.microsoft.com/wiki/contents/articles/13460.troubleshoot-cluster-aware-updating-cau-in-windows-server-2012-failover-clusters.aspx) (<http://social.technet.microsoft.com/wiki/contents/articles/13460.troubleshoot-cluster-aware-updating-cau-in-windows-server-2012-failover-clusters.aspx>).

Your feedback on any issues you encounter will be greatly appreciated. Feedbacks on SQL Server update failures should be posted through [SQL Server Connect Feedback Center](https://connect.microsoft.com/SQLServer/Feedback) (<https://connect.microsoft.com/SQLServer/Feedback>). Setup logs, detailed steps, and a clear description of the update failure environment are essential for us to investigate the issues you encountered.

# Best Practices

This section lists some best practices that are specific to updating instances of SQL Server using CAU. Review the topic [Requirements and Best Practices for Cluster-Aware Updating](http://technet.microsoft.com/library/jj134234.aspx) (http://technet.microsoft.com/library/jj134234.aspx) before reviewing the following list.

1. In order to prevent down-time on non-target SQL Server instances while patching target SQL Server instances, do not install multiple instances of SQL Server 2012 on the same set of nodes, especially on the primary node.
2. Because CAU drains the node before applying the update, make sure that the node used by the instance of SQL Server 2012 is not shared with an instance of SQL Server that is an older release that is installed on just one node, or on two nodes where one of the nodes may be in a maintenance state. This will lead to node drain failure because CAU is not able to find another node to failover the older instance of SQL Server.
3. If AlwaysOn availability groups exist on your cluster, make sure there are two replicas set to **automatic failover**. If you have a mix of standalone and clustered SQL Server 2012 failover instances and only one automatic failover replica, make sure that the AlwaysOn availability group primary replica and the SQL Server failover cluster instance primary node are on the same node. This will minimize downtime for AlwaysOn availability group during failover.

# Additional Resources

[CAU Scenario Overview on TechNet](http://technet.microsoft.com/library/hh831694.aspx) (http://technet.microsoft.com/library/hh831694.aspx)

[CAU Frequently Asked Questions](http://technet.microsoft.com/library/hh831367.aspx) (http://technet.microsoft.com/library/hh831367.aspx)

[CAU Troubleshooting](http://social.technet.microsoft.com/wiki/contents/articles/13460.troubleshoot-cluster-aware-updating-cau-in-windows-server-2012-failover-clusters.aspx) (http://social.technet.microsoft.com/wiki/contents/articles/13460.troubleshoot-cluster-aware-updating-cau-in-windows-server-2012-failover-clusters.aspx)

# Conclusion

In conclusion, I would like to highlight that this is the first time SQL Server update leverages CAU. As such, it is very important that you carefully review the CAU documentation and the information in this whitepaper before using CAU to update your SQL Server failover clustered instances. Please send your comments and feedback to help us develop future enhancements to the process of updating SQL Server failover clustered instances.

**For more information:**

<http://www.microsoft.com/sqlserver/>: SQL Server Web site

<http://technet.microsoft.com/en-us/sqlserver/>: SQL Server TechCenter

<http://msdn.microsoft.com/en-us/sqlserver/>: SQL Server DevCenter

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

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

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

[Send feedback](mailto:sqlfback@microsoft.com?subject=Patching%20SQL%20Server%20Failover%20Cluster%20Instances%20with%20Cluster-Aware%20Updating%20(CAU)).