Microsoft

Microsoft® System Center

Software Update Management Field Experience

Andre Della Monica, Chris Shilt, Russ Rimmerman, Rushi Faldu
Mitch Tulloch, Series Editor

Microsoft

Microsoft® System Center

Software Update Management Field Experience

Andre Della Monica, Chris Shilt, Russ Rimmerman, Rushi Faldu
Mitch Tulloch, Series Editor

PUBLISHED BY
Microsoft Press
A division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Library of Congress Control Number
ISBN: 978-0-7356-9584-9

Printed and bound in the United States of America.

First Printing

Microsoft Press books are available through booksellers and distributors worldwide. If you need support related to this book, email Microsoft Press Support at mspinput@microsoft.com. Please tell us what you think of this book at http://aka.ms/tellpress.

This book is provided "as-is" and expresses the author's views and opinions. The views, opinions and information expressed in this book, including URL and other Internet website references, may change without notice.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

Acquisitions Editor: Karen Szall
Developmental Editor: Karen Szall
Editorial Production: Megan Smith-Creed
Copyeditor: Megan Smith-Creed
Cover: Twist Creative • Seattle

Contents

This page intentionally left blank

Introduction

Ever since the advent of the Internet, security has been a concern for all users whose computers are vulnerable in the "biggest computer network in the free world." Because Windows is the most popular operating system for organizations and individual consumers, Microsoft has also always been particularly concerned with security.

To address security concerns, Microsoft first introduced the Windows Update capability with the launch of Windows 95. In the initial version (v3) of Windows Update, users had to manually visit the Windows Update website. An ActiveX control would then run on their computer and determine which software updates should be downloaded and installed on the user's computer. Windows 98 expanded this to include not only security updates but also optional features, driver updates, and desktop themes. Windows Update capability was also added to Windows NT 4.0.

Microsoft next released a tool called the Critical Update Notification Utility, which Windows 98 and Windows 2000 users could download from the Windows Update website and then use to download and install critical updates on their computers. The Critical Update Notification Utility was then discontinued and replaced with the Automatic Updates feature in Windows Millennium Edition (Me) and Windows 2000 Service Pack 4. Automatic Updates was designed to check for new updates every 24 hours and could automatically download and notify the user when they were ready to be installed on the computer. Other improvements to Windows Update were also made, for example the introduction of the Background Intelligent Transfer Service (BITS) in Windows 2000 SP3 and Windows XP.

In February 2005, Microsoft announced a beta release of Microsoft Update as an optional alternative to Windows Update for obtaining software updates for Windows and also for other Microsoft products. Several years later, Microsoft Office Update was introduced to enable updating of certain applications in the Microsoft Office suite. Beginning with Windows Vista and Windows Server 2008, the website download model was entirely replaced by a built-in user interface within Windows that allows updates to be selected and downloaded.

But in enterprise environments, software updates also need to be managed. Windows Server Update Services (WSUS), previously known as Software Update Services (SUS), was released in 2002, and Microsoft released a SUS Feature Pack add-on for their System Management Server (SMS) 2.0 product. The next version SMS 2003 included an Inventory Tool for Microsoft Updates (ITMU) that provided patch-management capability, although it was not fully integrated into the SMS product.

With the release of System Center Configuration Manager 2007, the Software Updates feature was completely re-written and integrated with WSUS. In general, distributing software updates through Configuration Manager using the WSUS engine works well. Feedback received by the Configuration Manager product team and Microsoft Customer Support

Services indicates that customers like the level of flexibility provided by this solution. However, feedback also shows that customers are often confused because there are too many ways to accomplish the same tasks.

In the current platform System Center 2012 R2 Configuration Manager, the Software Updates feature is quite mature and more robust than ever. The process for creating and maintaining updates has been improved, and the user interface is more self-explanatory, making it easier to create a group of updates to target collections of machines. From the server infrastructure perspective, there is much more functionality and flexibility, providing a reliable and seamless software updates process for corporate environments.

This book addresses some of the gaps and pain points you might encounter when implementing, administering, and troubleshooting Software Updates using Configuration Manager 2012 R2. We developed the topics for this book based on our experiences working as Premier Field Engineers and Microsoft Consultants in customer environments on a daily basis.

We hope you enjoy this book and our shared experiences from the field. May they help you build a stronger technical knowledge base so you can achieve your IT objectives.

Andre Della Monica

Premier Field Engineer, Microsoft Premier Services

About the companion content

The companion content for this book can be downloaded from the following page:

http://aka.ms/SUMFE/files

The companion content includes the following:

- In Chapter 1, two SQL query examples from the section titled "Software Update Point failover process"
- In Chapter 4, SQL Query 1, which provides a list of computers, total targeted updates, total installed, total required, % compliant, number of missing updates, and update status, and SQL Query 2, which can be used to create a report using SQL Reporting Services
- In Chapter 5, the sample Windows PowerShell scripts from the section titled "Automating software update database maintenance"

Acknowledgments

The Series Editor would like to thank the following individuals at Microsoft who reviewed the outlines for the proposed titles in this series and provided helpful feedback to the authors:

- David Ziembicki
- Adam Fazio
- Robert Larson
- David Stoker
- Joel Yoker

Free ebooks from Microsoft Press

From technical overviews to in-depth information on special topics, the free ebooks from Microsoft Press cover a wide range of topics. These ebooks are available in PDF, EPUB, and Mobi for Kindle formats, ready for you to download at:

http://aka.ms/mspressfree

Check back often to see what is new!

Errata, updates, & book support

We've made every effort to ensure the accuracy of this book and its companion content. You can access updates to this book—in the form of a list of submitted errata and their related corrections—at:

http://aka.ms/SUMFE/errata

If you discover an error that is not already listed, please submit it to us at the same page.

If you need additional support, email Microsoft Press Book Support at *mspinput@microsoft.com*.

Please note that product support for Microsoft software and hardware is not offered through the previous addresses. For help with Microsoft software or hardware, go to *http://support.microsoft.com*.

We want to hear from you

At Microsoft Press, your satisfaction is our top priority, and your feedback our most valuable asset. Please tell us what you think of this book at:

http://aka.ms/tellpress

The survey is short, and we read every one of your comments and ideas. Thanks in advance for your input!

Stay in touch

Let's keep the conversation going! We're on Twitter: *http://twitter.com/MicrosoftPress*.

CHAPTER 1

Understanding software update architecture: server side

Microsoft System Center 2012 Configuration Manager integrates client management in mobile, physical, and virtual environments. An important part of client management is the ability to implement a reliable patch-management process to keep organizations secured. One of the core features of Configuration Manager is Software Updates, which provides a consistent patch-management solution using existing Microsoft technologies. Rather than reinventing the wheel, the Configuration Manager product team incorporated Windows Server Update Services (WSUS) in Configuration Manager as the main engine used by Software Updates. This feature has been further enhanced in Configuration Manager 2012. This chapter provides an overview of Software Updates from the server perspective.

Before implementing this feature, it is highly recommended that you become familiar with the various steps of the process:

- Fundamentals
- The synchronization process
- Software Updates policy deployment

Fundamentals

Before exploring Software Updates from the server perspective, it is helpful to understand the key elements, principles, and data workflow of Software Updates.

Configuration items

Configuration items are very important elements in Configuration Manager because they are what describes a configuration and what you use to manipulate or evaluate updates. For example, a software update can be required, not required, or it can have multiple compliance states such as installed, not installed, unknown, and so on. The main engine of the configuration item is the compliance settings feature, but other features such as Operating

System Deployment (OSD) and Application Management take advantage of configuration items. The focus of this book is the Software Updates feature, which includes the following Software Updates configuration item types:

- **Update source files** These are the .msp, .cab, and executable files, which need to be evaluated to determine where the content is. This is used by the distribution point to have a reference of its content.
- **Individual software updates** The status of a software update (required, not required, installed, not installed, and so on) can vary, depending on the software update evaluation performed in the client side.
- **Software update groups** When you do routine maintenance, such as removing expired software updates or moving updates between software update groups, the compliance status of the software update group changes to Unknown.

Software update point

A software update point is the Configuration Manager site system role that you must install before you can implement software updates, and it has only one prerequisite: WSUS. Configuration Manager integrates some functionalities of WSUS, such as update catalog download and distribution capabilities, but it uses its own functionality to deploy and install the updates. So, in general, when installing the software update point, you scan with WSUS but install the updates with Configuration Manager.

When thinking about capacity and planning for a software update point, it is important to note that when you implement this role on a system that co-exists with another Configuration Manager site system, it is possible to support up to 25,000 clients, but if you are installing software updates on a server with no other Configuration Manager site system role, it is possible to support up to 100,000 clients. In addition, there is a practical limit of 1,000 update objects per deployment package when you are working with manual software update deployments and when you are using automated deployment rules (ADR), which will be discussed in Chapter 5, "Software updates automation."

Multiple software update points

In Configuration Manager 2012 SP1, the software update point was completely redesigned to allow you to add multiple software update point site systems to a Configuration Manager primary site. In addition, you can configure the software update points in the same forest, in a cross-forest, or for Internet-based clients. When you install more than one software update point for one Configuration Manager site, clients automatically connect to the most appropriate one based on their forest boundary. This behavior provides redundancy without requiring a Network Load Balancing (NLB) cluster, which is still possible to have when using the Configuration Manager SDK or Windows PowerShell. Remember that you cannot install more than one software update point on a Configuration Manager secondary site.

IMPORTANT The active software update point concept no longer exists in System Center 2012 R2 Configuration Manager.

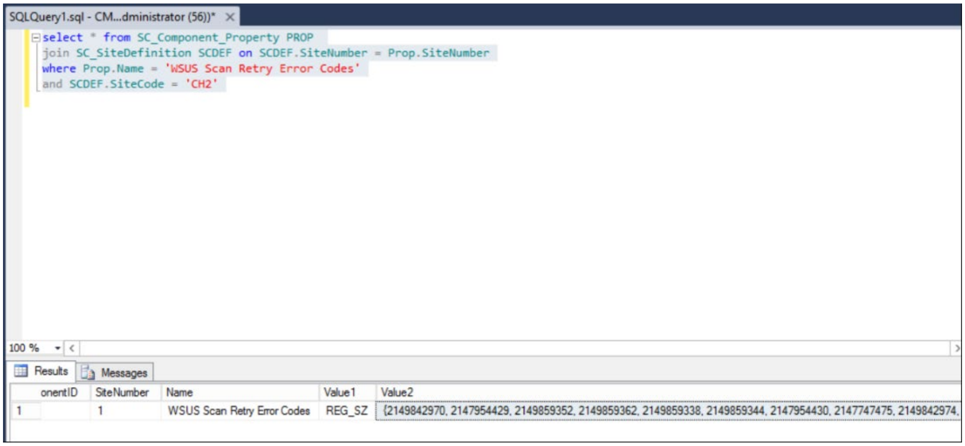
Software update point failover process

The software update point failover process is not as robust as the NLB for load balancing. When installed for the first time, assuming that your environment has the Software Updates feature enabled from the Configuration Manager client settings, a client randomly selects an available software update point, with which it will maintain affinity until it is no longer able to communicate with that software update point. A combination of four failed retries at 30-minute intervals and non-retry error codes is what determines that the client is no longer able to communicate with the software update point and causes the switch to another software update point. It is important to note that, by default, the software update point only retries on 36 retry error codes, which in part are used by Windows Update Agent (WUA) and WINHTTP to determine that a scan has failed. If the error code that a particular client is receiving is not in this list, the failover does not happen.

Based on interactions with Microsoft customers, the best practice for supporting the software update point failover is to run the following SQL query against the Configuration Manager database to gather all 36 retry error codes. Note that you should replace the site code “CM2” with your site code.

```
select * from SC_Component_Property PROP
join SC_SiteDefinition SCDEF on SCDEF.SiteNumber = Prop.SiteNumber
where Prop.Name = 'WSUS Scan Retry Error Codes'
and SCDEF.SiteCode = 'CM2'
```

In the SQL query output, the Value2 field contains the WSUS retry codes, as shown in Figure 1-1.



```
select * from SC_Component_Property PROP
join SC_SiteDefinition SCDEF on SCDEF.SiteNumber = Prop.SiteNumber
where Prop.Name = 'WSUS Scan Retry Error Codes'
and SCDEF.SiteCode = 'CH2'
```

	onentID	SiteNumber	Name	Value1	Value2
1		1	WSUS Scan Retry Error Codes	REG_SZ	{2149842970, 2147954429, 2149859352, 2149859362, 2149859338, 2149859344, 2147954430, 2147747475, 2149842974,

FIGURE 1-1 SQL query statement to retrieve the WSUS scan retry error codes

The WSUS retry error code list can be retrieved using Windows Management Instrumentation (WMI) by launching the WMI utility WBEMTEST and running the following WMI query language (WQL), where PR1 is your Configuration Manager primary site code.

```
root\SMS\site_PR1
Query - select * from SMS_SCI_SCProperty where propertyname like '%scan retry error
codes%'
SMS_WSUS_CONFIGURATION_MANAGER -> WSUS Scan Retry Error Codes
```

If a client is failing with a very common error code, and if that error is not in the WSUS retry error list, then the client will never fail over to the other software update point. For example, the error 0x80072ee2 is a common network timeout error, and if a client is failing with that error, it would never be able to scan against the software update point until the issue is fixed.

To work around this issue, you can add the common error to the WSUS retry error code list inside WMI on the Configuration Manager primary site. To do so, complete the following steps:

1. Run WBEMTEST with your administrator account.
2. Connect to root\sms\site_<sitecode>.
3. Click Query, and run the following query:

   ```
   select * from sms_sci_component where
   componentname='SMS_WSUS_CONFIGURATION_MANAGER'
   ```

4. Double-click the object.
5. Double-click the Props properties.
6. Click View Embedded.
7. Double-click the query result and locate the PropertyName WSUS Scan Retry Error Codes.
8. Double-click Value2. Add the error code to the list, for example, 2147954402.
9. Click Save Property.
10. Click Save Object.
11. Click Close, and then click Save Property.
12. Click Save Object.

IMPORTANT **In order to automate these steps and accomplish the same changes, you can leverage the Configuration Manager 2012 R2 SDK.**

Internet-based software update point

An Internet-based software update point accepts communication from devices on the Internet. This role must be assigned to a site system that is remote from the site server, located in a perimeter network, and accessible to Internet-based devices. The Internet-based software update point synchronizes with a software update point in the same Configuration Manager site. When the connectivity between the software update point and Internet-based software update point is limited, you can manually synchronize software updates by using the export and import process.

Software updates on a secondary site

A software update point is optional on a secondary site. When a software update point is installed on a secondary site, the WSUS database is configured as a replica instead of as an autonomous WSUS instance, which is how the WSUS database is configured when installing the software update point on a primary site or central administration site.

Devices assigned to a secondary site are configured to use a software update point at the parent site when a software update point is not configured at the secondary site. Typically, a software update point is installed at a secondary site when there is limited network bandwidth between devices assigned to the secondary site and the software update point at the parent site or when the software update point is approaching capacity. After the software update point is successfully installed and configured on the secondary site, Local Group Policy is updated on client computers, and they start using the new software update point.

> **IMPORTANT** **Internet-based software update points are not supported on Configuration Manager secondary sites.**

Using an existing WSUS server

At the top-level System Center 2012 R2 Configuration Manager site, an existing WSUS server can be configured as the upstream data source location. During the synchronization process, the site connects to this location to synchronize software updates. For example, if there is an existing WSUS server that is not part of the Configuration Manager hierarchy, this server can be specified as the existing WSUS server to synchronize software updates. That helps organizations that need to meet security requirements, such as not allowing all servers to have access to the Internet, which has become a common requirement in today’s IT world since there are many threats coming from the Internet.

Software update data

An individual software update consists of two pieces:

- The metadata
- The binary file

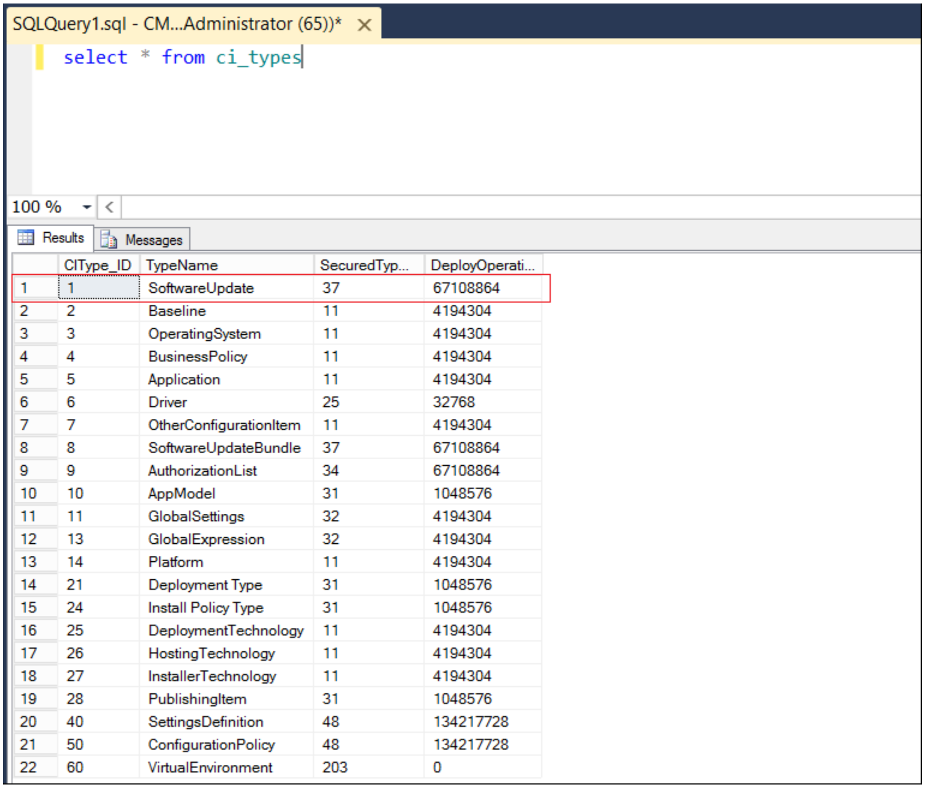
The metadata provides information about an update, such as the description, product supported, applicability rules, classification, article ID, file hash information, command line, and even the URL containing the location for downloading a particular update. If you want to see the details about an update metadata file, you can run a few queries against the Configuration

Manager database after you have synchronized the software update catalog from the Microsoft cloud.

Remember that an update metadata file is also a configuration item in the Configuration Manager database; therefore, it is possible to get more details about a particular update. The first step is to find the software update configuration item number. Gather this information by running the following SQL query against the Configuration Manager database:

```
select * from ci_types
```

The output for this SQL query returns a list of all configuration items as shown in Figure 1-2.



```
select * from ci_types
```

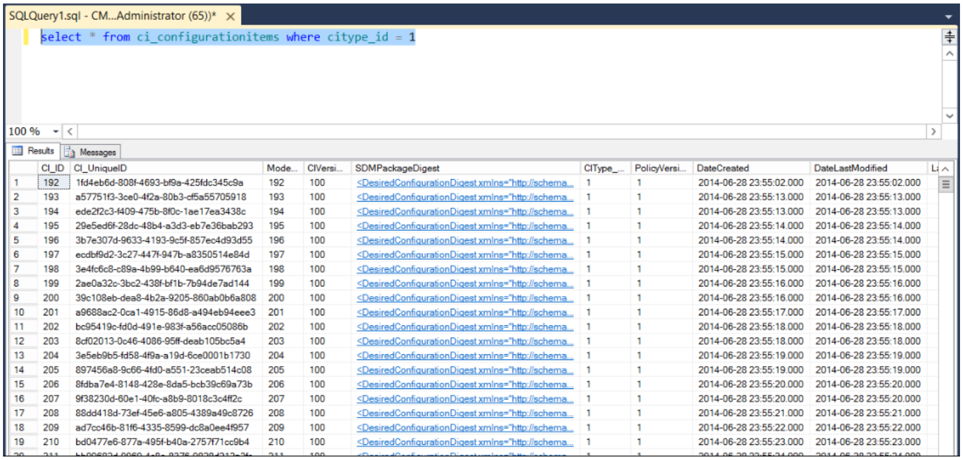
	CIType_ID	TypeName	SecuredTyp...	DeployOperati...
1	1	SoftwareUpdate	37	67108864
2	2	Baseline	11	4194304
3	3	OperatingSystem	11	4194304
4	4	BusinessPolicy	11	4194304
5	5	Application	11	4194304
6	6	Driver	25	32768
7	7	OtherConfigurationItem	11	4194304
8	8	SoftwareUpdateBundle	37	67108864
9	9	AuthorizationList	34	67108864
10	10	AppModel	31	1048576
11	11	GlobalSettings	32	4194304
12	13	GlobalExpression	32	4194304
13	14	Platform	11	4194304
14	21	Deployment Type	31	1048576
15	24	Install Policy Type	31	1048576
16	25	DeploymentTechnology	11	4194304
17	26	HostingTechnology	11	4194304
18	27	InstallerTechnology	11	4194304
19	28	PublishingItem	31	1048576
20	40	SettingsDefinition	48	134217728
21	50	ConfigurationPolicy	48	134217728
22	60	VirtualEnvironment	203	0

FIGURE 1-2 The SQL query statement to retrieve the configuration item types

The second step is to run the following SQL query against the Configuration Manager database to see all of the software update configuration items:

```
select * from ci_configurationitems where citype_id = 1
```

The output for the SQL query returns all the software update configuration items as shown in Figure 1-3.

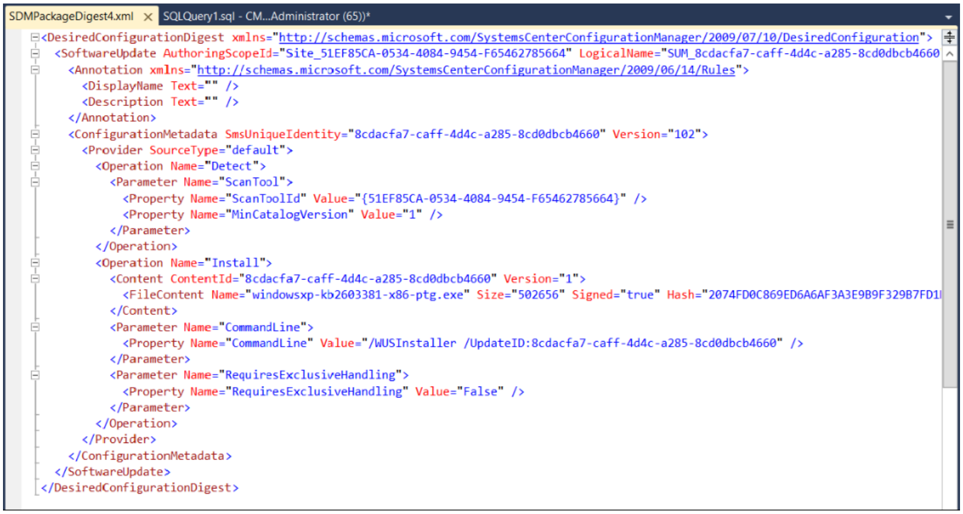


```sql
select * from ci_configurationitems where citype_id = 1
```

	CI_ID	CI_UniqueID	Mode...	CIVersi...	SDMPackageDigest	CIType_...	PolicyVersi...	DateCreated	DateLastModified	Li
1	192	1fd4eb6d-808f-4693-bf9a-425fdc345c9a	192	100	<DesiredConfigurationDigest xmlns="http://schema...	1	1	2014-06-28 23:55:02.000	2014-06-28 23:55:02.000	
2	193	a57751f3-3ce0-4f2a-80b3-cf5a55705918	193	100	<DesiredConfigurationDigest xmlns="http://schema...	1	1	2014-06-28 23:55:13.000	2014-06-28 23:55:13.000	
3	194	ede2f2c3-f409-475b-8f0c-1ae17ea3438c	194	100	<DesiredConfigurationDigest xmlns="http://schema...	1	1	2014-06-28 23:55:13.000	2014-06-28 23:55:13.000	
4	195	29e5ed6f-28dc-48b4-a3d3-eb7e36bab293	195	100	<DesiredConfigurationDigest xmlns="http://schema...	1	1	2014-06-28 23:55:14.000	2014-06-28 23:55:14.000	
5	196	3b7e307d-9633-4193-9c5f-857ec4d93d55	196	100	<DesiredConfigurationDigest xmlns="http://schema...	1	1	2014-06-28 23:55:14.000	2014-06-28 23:55:14.000	
6	197	ecdbf9d2-3c27-447f-947b-a8350514e84d	197	100	<DesiredConfigurationDigest xmlns="http://schema...	1	1	2014-06-28 23:55:15.000	2014-06-28 23:55:15.000	
7	198	3e4fc6c8-c89a-4b99-b640-ea6d9576763a	198	100	<DesiredConfigurationDigest xmlns="http://schema...	1	1	2014-06-28 23:55:15.000	2014-06-28 23:55:15.000	
8	199	2ae0a32c-3bc2-438f-bf1b-7b94de7ad144	199	100	<DesiredConfigurationDigest xmlns="http://schema...	1	1	2014-06-28 23:55:16.000	2014-06-28 23:55:16.000	
9	200	39c108eb-dea8-4b2a-9205-860ab0b6a808	200	100	<DesiredConfigurationDigest xmlns="http://schema...	1	1	2014-06-28 23:55:16.000	2014-06-28 23:55:16.000	
10	201	a9688ac2-0ca1-4915-86d8-a494eb94eee3	201	100	<DesiredConfigurationDigest xmlns="http://schema...	1	1	2014-06-28 23:55:17.000	2014-06-28 23:55:17.000	
11	202	bc95419c-fd0d-491e-983f-a56acc05086b	202	100	<DesiredConfigurationDigest xmlns="http://schema...	1	1	2014-06-28 23:55:18.000	2014-06-28 23:55:18.000	
12	203	8cf02013-0c46-4086-95ff-deab105bc5a4	203	100	<DesiredConfigurationDigest xmlns="http://schema...	1	1	2014-06-28 23:55:18.000	2014-06-28 23:55:18.000	
13	204	3e5eb9b5-fd58-4f9a-a19d-6ce0001b1730	204	100	<DesiredConfigurationDigest xmlns="http://schema...	1	1	2014-06-28 23:55:19.000	2014-06-28 23:55:19.000	
14	205	897456a8-9c66-4fd0-a551-23ceab514c08	205	100	<DesiredConfigurationDigest xmlns="http://schema...	1	1	2014-06-28 23:55:19.000	2014-06-28 23:55:19.000	
15	206	8fdba7e4-8148-428e-8da5-bcb39c69a73b	206	100	<DesiredConfigurationDigest xmlns="http://schema...	1	1	2014-06-28 23:55:20.000	2014-06-28 23:55:20.000	
16	207	9f38230d-60e1-40fc-a8b9-8018c3c4ff2c	207	100	<DesiredConfigurationDigest xmlns="http://schema...	1	1	2014-06-28 23:55:20.000	2014-06-28 23:55:20.000	
17	208	88dd418d-73ef-45e6-a805-4389a49c8726	208	100	<DesiredConfigurationDigest xmlns="http://schema...	1	1	2014-06-28 23:55:21.000	2014-06-28 23:55:21.000	
18	209	ad7cc46b-81f6-4335-8599-dc8a0ee4f957	209	100	<DesiredConfigurationDigest xmlns="http://schema...	1	1	2014-06-28 23:55:22.000	2014-06-28 23:55:22.000	
19	210	bd0477e6-877a-495f-b40a-2757f71cc9b4	210	100	<DesiredConfigurationDigest xmlns="http://schema...	1	1	2014-06-28 23:55:23.000	2014-06-28 23:55:23.000	

FIGURE 1-3 The SQL query statement to retrieve all software updates configuration items

After the SQL query output appears, click the update link in the SDMPackageDigest column to get more details about a specific software update configuration item, as shown in Figure 1-4.



```xml
<DesiredConfigurationDigest xmlns="http://schemas.microsoft.com/SystemsCenterConfigurationManager/2009/07/10/DesiredConfiguration">
  <SoftwareUpdate AuthoringScopeId="Site_51EF85CA-0534-4084-9454-F65462785664" LogicalName="SUM_8cdacfa7-caff-4d4c-a285-8cd0dbcb4660
    <Annotation xmlns="http://schemas.microsoft.com/SystemsCenterConfigurationManager/2009/06/14/Rules">
      <DisplayName Text="" />
      <Description Text="" />
    </Annotation>
    <ConfigurationMetadata SmsUniqueIdentity="8cdacfa7-caff-4d4c-a285-8cd0dbcb4660" Version="102">
      <Provider SourceType="default">
        <Operation Name="Detect">
          <Parameter Name="ScanTool">
            <Property Name="ScanToolId" Value="{51EF85CA-0534-4084-9454-F65462785664}" />
            <Property Name="MinCatalogVersion" Value="1" />
          </Parameter>
        </Operation>
        <Operation Name="Install">
          <Content ContentId="8cdacfa7-caff-4d4c-a285-8cd0dbcb4660" Version="1">
            <FileContent Name="windowsxp-kb2603381-x86-ptg.exe" Size="502656" Signed="true" Hash="2074FD0C869ED6A6AF3A3E9B9F329B7FD1
          </Content>
          <Parameter Name="CommandLine">
            <Property Name="CommandLine" Value="/WUSInstaller /UpdateID:8cdacfa7-caff-4d4c-a285-8cd0dbcb4660" />
          </Parameter>
          <Parameter Name="RequiresExclusiveHandling">
            <Property Name="RequiresExclusiveHandling" Value="False" />
          </Parameter>
        </Operation>
      </Provider>
    </ConfigurationMetadata>
  </SoftwareUpdate>
</DesiredConfigurationDigest>
```

FIGURE 1-4 SQL query statement showing an example of an individual software update after clicking on SDMPackageDigest column link

The second piece of the software update file, the binary file, is the actual executable (.exe) file or Windows Installer (.msi) file that the Configuration Manager client downloads from the Configuration Manager distribution point to its cache when it is necessary to install a software update.

IMPORTANT It is worth reinforcing that when the client downloads an update metadata file during a scanning process (discussed in Chapter 2, "Understanding software update architecture: client side"), it communicates with the software update point and the WSUS server. When the client downloads the actual binary files to install an update, it communicates directly with Configuration Manager by downloading the binary files from the distribution point.

The synchronization process

All software update points and WSUS instances will synchronize updates within a Configuration Manager hierarchy from a configured source according to a pre-defined schedule or even a manual start by an administrator. The goal of this process is for the software update point to download the Microsoft Update Catalog from the Internet and transfer it to the Configuration Manager database. For a top-level site such as the central administration site, the synchronization process happens directly with the Microsoft Update Catalog hosted in the Microsoft cloud through an Internet connection. As for the child primary sites, and even the secondary sites, this communication happens through the Configuration Manager Database Replication Service (DRS), the new mechanism for Configuration Manager site-to-site replication as of System Center 2012 Configuration Manager. In the case of a single primary site, the primary site communicates through an Internet connection to download the Microsoft Update Catalog.

As previously discussed in this chapter, you can also leverage an existing WSUS server to synchronize the Microsoft Update Catalog if security restrictions prevent Internet access by the other servers inside your organization.

Scheduled vs. manual synchronization

There are two methods for synchronizing updates from the Microsoft Update Catalog with Configuration Manager:

- **Schedule initiated synchronization** Scheduled synchronization can be configured only from the top-level site. To set up scheduled synchronization, navigate to the Administration workspace, expand Site Configuration, click Sites, select the top-level site from your hierarchy, click Configuration Site Components, and then select Software Update Point. The configuration can be changed on the Synch Schedule tab. When Configuration Manager performs a scheduled synchronization, it performs a full synchronization from the Microsoft Update Catalog. This method repairs any issues you might have had with the previous synchronization cycle.

  In addition, this method allows you to synchronize the expired, superseded, and declined updates with the Microsoft Update Catalog, which doesn’t happen when you use the manually initiated synchronization method. There are a couple of scenarios where a full synchronization will be triggered automatically by Configuration Manager:

- The timeout countdown setting inside the Site Control file is passed.
- The superseded rules are changed.
- The updates categories are changed in the Configuration Manager console.

- **Manually initiated synchronization** Manually initiated synchronization can be initiated only from the top-level site within a Configuration Manager hierarchy. It is also known as the delta synchronization because it synchronizes only the updates that have been changed since the last synchronization. The last synchronization time is stored in the Site Control file, which is stored in the Configuration Manager database as of Configuration Manager 2012 RTM.

Field experience

You can manipulate either a full or a delta synchronization in Configuration Manager by creating an empty flag file inside the .\inboxes\wsyncmgr.box, located in the Configuration Manager installation path. For a full synchronization, you need to name the flag file FULL.SYN. To trigger a delta synchronization, you need to name the flag file SELF.SYN. You can review the whole synchronization process by viewing the WSYNCMGR.log, which is located in the Configuration Manager site server installation path\Logs folder.

IMPORTANT **It is not recommended that you manipulate the synchronization every time you need to synchronize your updates catalog, but it is definitely a useful option when troubleshooting the software updates synchronization process.**

The software update point features

The software update point communicates with WSUS in three ways:

- **WSUS Control Manager** This feature calls the WSUS APIs that are needed during the software update point and WSUS communications and verifies that the correct versions of the APIs are being used. In addition, the WSUS Control Manager checks the connection to WSUS by connecting to the APIRemoting30 virtual directory in Internet Information Services (IIS). You can check whether the connection succeeded or failed in WSUSCTRL.log, located in the Configuration Manager installation path\Logs folder.
- **WSUS Configuration Manager** This feature manages the software update point configuration definitions such as classifications, language, products, and others to synchronize the metadata. It makes a call to the APIRemoting30 virtual directory in IIS to set all the software update point configuration definitions in WSUS. There is no need to configure this in the WSUS installation wizard; during the WSUS setup, you can skip it, and work on these configurations later by using the Configuration Manager console.

- **WSUS Synchronization Manager** This feature is responsible for synchronizing its updates with the Microsoft Update Catalog over the Internet. The software update point calls the WSUS APIs, and then WSUS starts the Internet connection to synchronize its updates, saving these updates first into the WSUS database. The second step of this process is to synchronize the WSUS database tables with the Configuration Manager site database tables by converting all the updates into configuration items, as previously explained in this chapter.

The metadata synchronization

Figure 1-5 shows an overview of the synchronization process. The details are as follows:

1. Software update classes and products are selected for synchronization, and the synchronization schedule is configured or manually initiated. WSUS Synchronization Manager on the site server calls an API that requests that the WSUS server initiates a synchronization to Microsoft Update. WSM continues to poll the WSUS server until synchronization completes.
2. The WSUS server requests the software update metadata from Microsoft Update. Microsoft Update returns the software update metadata to the WSUS server. The WSUS server stores the metadata in the WSUS database.
3. WSM polls the WSUS server, detects that WSUS synchronization has successfully completed, requests the software update metadata from the WSUS server, and inserts it into the Configuration Manager site database. If the update already exists, it is not inserted. If it has changed or, for example, it is expired, the associated attribute is modified. This is the point where the software updates are converted into configuration items.
4. The site server updates the software update version information in the machine policy and copies it to the management point so that the next time the clients scan for updates, they scan against the most recent Microsoft Update Catalog.

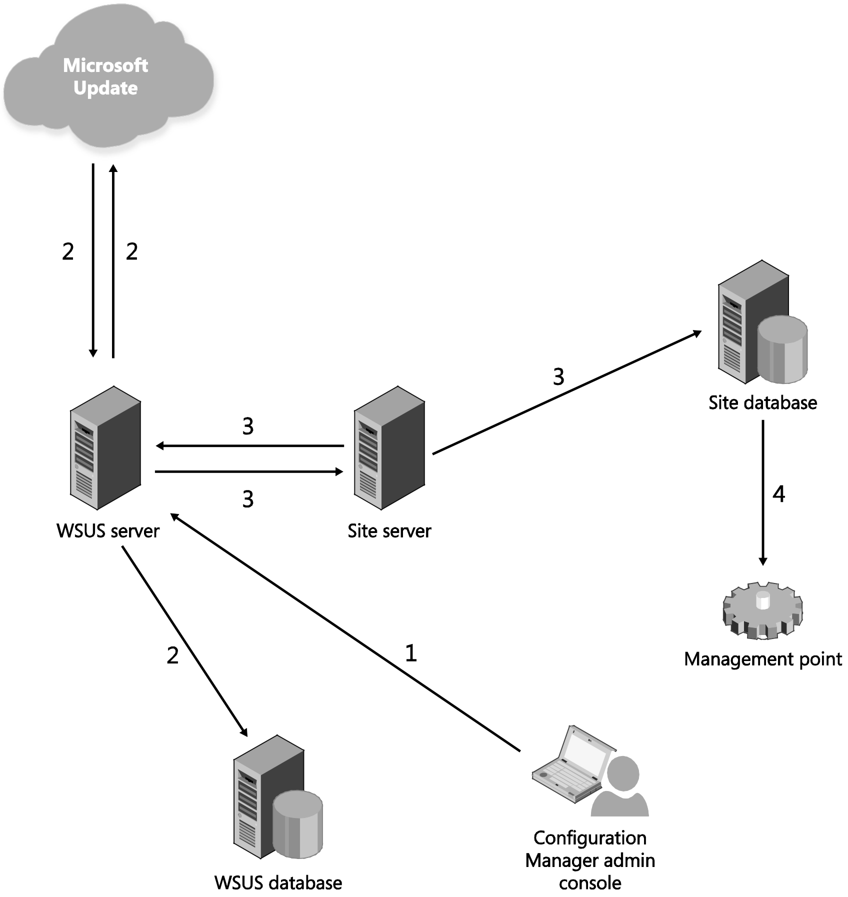


FIGURE 1-5 An overview of the software update metadata synchronization process

Configuration Manager inter-site replication

After the metadata synchronization is finished, there is inter-site replication between the Configuration Manager sites. The various details behind the Configuration Manager inter-site replication is beyond the scope of this book, but basically, when you have a Configuration Manager hierarchy with a central administration site, primary sites, and secondary sites, the software updates and the software update groups, both metadata, will be replicated through the Configuration Manager SQL replication mechanism known as database replication services, whether they were created in the central administration site or a child primary site, because they are considered global data within the Configuration Manager hierarchy.

All the inter-site communication happens through port 4022, using SQL service broker protocol, and port 1433, which Microsoft SQL Server uses to make sure a connection can be established between the SQL databases.

Firewall considerations

If an organization does not allow the HTTP ports 80 or 443 to be open through the firewall, you can restrict the access to the following domains so that the software update point can communicate with the Microsoft Update Catalog over the Internet:

- http://windowsupdate.microsoft.com
- http://*.windowsupdate.microsoft.com
- https://*.windowsupdate.microsoft.com
- http://*.update.microsoft.com
- https://*.update.microsoft.com
- http://*.windowsupdate.com
- http://download.windowsupdate.com
- http://download.microsoft.com
- http://*.download.windowsupdate.com
- http://wustat.windows.com
- http://ntservicepack.microsoft.com

> **IMPORTANT** The steps for configuring the firewall are meant for a corporate firewall positioned between the software update point and the Internet. Since the software update point initiates all of its network traffic, there is no need to configure the Windows Firewall on the software update point server.

The flow of binary data

When deploying a software update group or even when manually downloading a group of updates by using the download option from the Configuration Manager console, it is necessary to associate these downloaded updates with a software updates deployment package, which is discussed in Chapter 3, "Managing software updates." The deployment package is like a bucket and contains the binary files that you need to distribute to a distribution point to make them available for the Configuration Manager clients. Figure 1-6 shows the flow of binary data. Here are the details:

1. The administrator creates a new software updates deployment (or downloads the updates before deploying the software updates) and associates it with a software update deployment package.
2. The Configuration Manager site server requests the software updates binary files from the source location defined in the software update deployment. This can be from Microsoft Update or from a local source.

3. The software update binary files are stored temporarily in a folder on the site server.
4. The Configuration Manager site server copies the software update binary files to the content library on the distribution point.

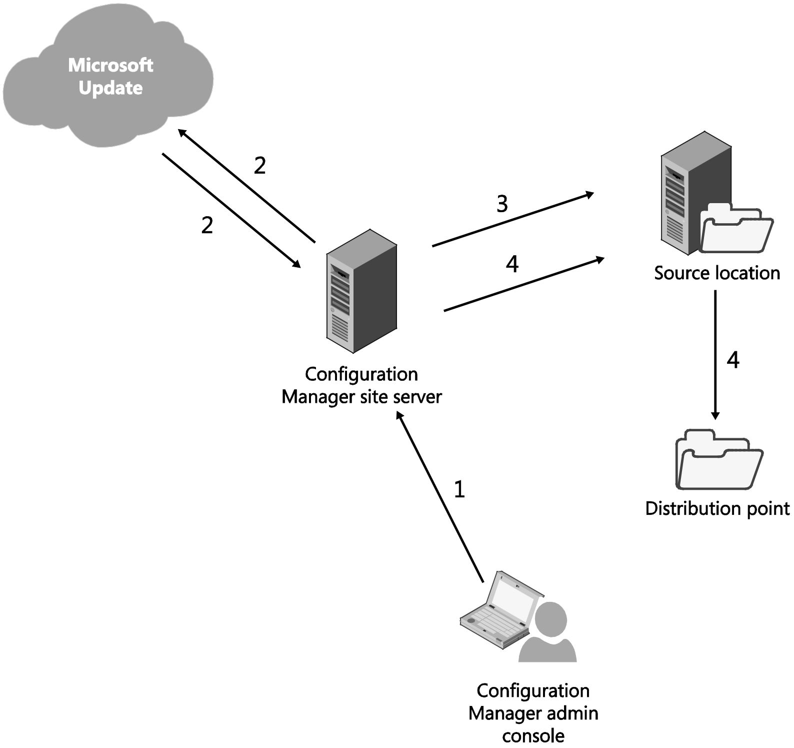

Microsoft Update
2
2
3
4
Source location
4
Configuration Manager site server
1
Distribution point
Configuration Manager admin console

FIGURE 1-6 An overview of the software update binary data flow process

The Configuration Manager site server uses the Software Update Patch downloader to download the software update binary data. You can check the download actions in PatchDownloader.log. This log can be found in c:\users\%username%\appdata\local\temp when running the Configuration Manager console from a terminal server section, or in the Configuration Manager site server installation path \SMS_CCM\Logs folder.

IMPORTANT **By default, the user account that is logged as performing the software update actions in the Configuration Manager console is the one used to download the software update binary data. You can use a software update point proxy server account to perform this operation. For more information, see the article at *http://technet.microsoft.com/en-us/library/hh427337.aspx*.**

Software Update policy deployment

The main goal for installing the software update point and synchronizing the updates with the Microsoft Update Catalog is to be able to make software update policy available for the Configuration Manager clients. It is important to understand that the act of synchronizing updates, or even creating software update groups by using the Configuration Manager console, does not create any software update policy. Policy Provider is responsible for creating policy, not only in the context of software updates, but for all other Configuration Manager features such as operating system deployment, application deployment, and others. The features involved in policy creation are:

- SMS Provider
- SMS Database monitoring
- Distribution Manager
- Offer Manager
- Policy Provider

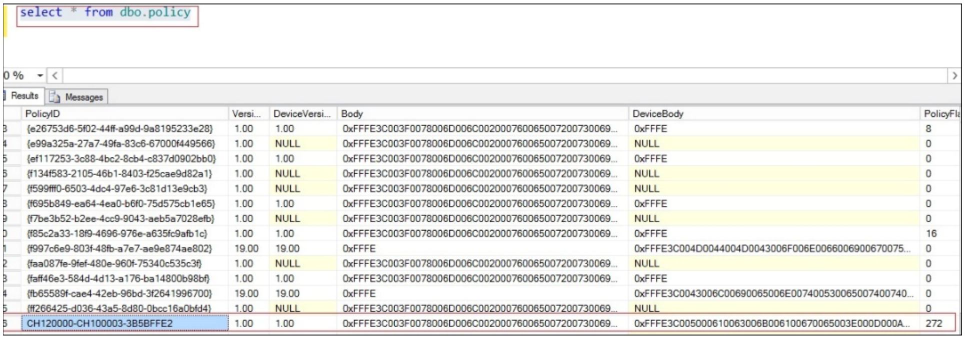
There is a lot happening behind the scenes when you simply create a software update group and deploy it to a group of servers or workstations. The flow described in the next section shows all of the features working together to prepare the software update policy so that the clients can download and run the updates when communicating with the management point later.

The policy creation flow

The following steps describe how to the policy is created:

1. The administrator creates the software update group and deploys it. The SMS provider and SMS Database Monitor are responsible for committing the software update deployment details into the WMI and SQL database. These activities are registered in SMSProv.log and SMSDBMon.log, both located in the Configuration Manager installation path\Logs folder.
2. The policy creation process is triggered by Distribution Manager, which creates an instruction file (.ofn) for Offer Manager. This is done because the process of creating a deployment requires a package to be created with its binary files, which are stored in a distribution point so the clients can locate, download, and run the deployment. Because the deployment instruction does not apply to every client, the Offer Manager is also responsible for applying the instruction for those clients pre-defined in the deployment settings. These activities are registered in DistMgr.log and OfferMgr.log, both located in the Configuration Manager installation path\Logs folder.
3. Policy Provider receives the .ofn instruction file from Offer Manager and processes and stores it in the Configuration Manager SQL database. All of the activities are registered in PolicyPv.log located in the Configuration Manager installation path\Logs folder.

With the policy officially created, you can verify it in the Configuration Manager SQL database. There are many tables that list policies. The two most important tables are Dbo.Policy and Dbo.PolicyAssignment. For example, in the Dbo.Policy table, the policy is stored by ID, as shown in Figure 1-7.

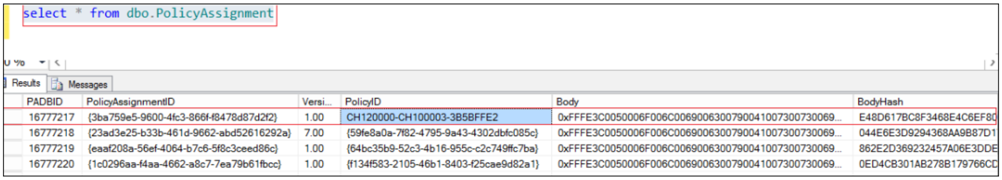


```
select * from dbo.policy
```

	PolicyID	Versi...	DeviceVersi...	Body	DeviceBody	PolicyFla
3	{e26753d6-5f02-44ff-a99d-9a8195233e28}	1.00	1.00	0xFFFE3C003F0078006D006C002000760065007200730069...	0xFFFE	8
4	{e99a325a-27a7-49fa-83c6-67000f449566}	1.00	NULL	0xFFFE3C003F0078006D006C002000760065007200730069...	NULL	0
5	{ef117253-3c88-4bc2-8cb4-c837d0902bb0}	1.00	1.00	0xFFFE3C003F0078006D006C002000760065007200730069...	0xFFFE	0
6	{f134f583-2105-46b1-8403-f25cae9d82a1}	1.00	NULL	0xFFFE3C003F0078006D006C002000760065007200730069...	NULL	0
7	{f599fff0-6503-4dc4-97e6-3c81d13e9cb3}	1.00	NULL	0xFFFE3C003F0078006D006C002000760065007200730069...	NULL	0
8	{f695b849-ea64-4ea0-b6f0-75d575cb1e65}	1.00	1.00	0xFFFE3C003F0078006D006C002000760065007200730069...	0xFFFE	0
9	{f7be3b52-b2ee-4cc9-9043-aeb5a7028efb}	1.00	NULL	0xFFFE3C003F0078006D006C002000760065007200730069...	NULL	0
0	{f85c2a33-18f9-4696-976e-a635fc9afb1c}	1.00	1.00	0xFFFE3C003F0078006D006C002000760065007200730069...	0xFFFE	16
1	{f997c6e9-803f-48fb-a7e7-ae9e874ae802}	19.00	19.00	0xFFFE	0xFFFE3C004D0044004D0043006F006E0066006900670075...	0
2	{faa087fe-9fef-480e-960f-75340c535c3f}	1.00	NULL	0xFFFE3C003F0078006D006C002000760065007200730069...	NULL	0
3	{faff46e3-584d-4d13-a176-ba14800b98bf}	1.00	1.00	0xFFFE3C003F0078006D006C002000760065007200730069...	0xFFFE	0
4	{fb65589f-cae4-42eb-96bd-3f2641996700}	19.00	19.00	0xFFFE	0xFFFE3C0043006C00690065006E007400530065007400740...	0
5	{ff266425-d036-43a5-8d80-0bcc16a0bfd4}	1.00	NULL	0xFFFE3C003F0078006D006C002000760065007200730069...	NULL	0
6	CH120000-CH100003-3B5BFFE2	1.00	1.00	0xFFFE3C003F0078006D006C002000760065007200730069...	0xFFFE3C005000610063006B006100670065003E000D000A...	272

FIGURE 1-7 The SQL query statement to retrieve all policies generated by the act of creating a deployment, such as software updates or other kind of deployments

The Dbo.PolicyAssignment shows a deployment being assigned to a collection of Configuration Manager clients, as shown in Figure 1-8.



```
select * from dbo.PolicyAssignment
```

PADBID	PolicyAssignmentID	Versi...	PolicyID	Body	BodyHash
16777217	{3ba759e5-9600-4fc3-866f-f8478d87d2f2}	1.00	CH120000-CH100003-3B5BFFE2	0xFFFE3C0050006F006C0069006300790041007300730069...	E48D617BC8F3468E4C6EF80
16777218	{23ad3e25-b33b-461d-9662-abd52616292a}	7.00	{59fe8a0a-7f82-4795-9a43-4302dbfc085c}	0xFFFE3C0050006F006C0069006300790041007300730069...	044E6E3D9294368AA9B87D1
16777219	{eaaf208a-56ef-4064-b7c6-5f8c3ceed86c}	1.00	{64bc35b9-52c3-4b16-955c-c2c749ffc7ba}	0xFFFE3C0050006F006C0069006300790041007300730069...	862E2D369232457A06E3DDE
16777220	{1c0296aa-f4aa-4662-a8c7-7ea79b61fbcc}	1.00	{f134f583-2105-46b1-8403-f25cae9d82a1}	0xFFFE3C0050006F006C0069006300790041007300730069...	0ED4CB301AB278B179766CD

FIGURE 1-8 The SQL query statement to retrieve all policies assigned to the Configuration Manager clients

IMPORTANT It is possible to review all of the policy ID references in the Configuration Manager site server logs, such as PolicyPv.log located in the configuration installation path \Logs folder as well as in the client logs after the policies are deployed to the client machines.

This page intentionally left blank

CHAPTER 2

Understanding software update architecture: client side

Microsoft System Center 2012 R2 Configuration Manager uses a client/server model that partitions tasks and workloads between providers (site servers and site systems) and service requesters (clients). The software update feature of Configuration Manager uses the same model by partitioning its tasks between the software update point server and the software update clients.

This chapter provides an overview of the software update client architecture responsible for keeping clients up to date. The chapter also describes the mechanisms used to scan, install, and report compliance data to the software update point server. A thorough understanding of how the various architecture features work together is essential to successfully implement, administer, and troubleshoot your software update environment.

Software update client architecture features

Several client features are involved in deploying software updates using Configuration Manager. It's important to be familiar with these client features because they can help you troubleshoot the flow of compliance data from the client to the server when something goes wrong. The primary software update client features are as follows:

- Windows Update Agent (WUA)
- Windows Update data store
- Configuration Manager Software Updates Client Agent
- Windows Management Instrumentation (WMI)
- Configuration Manager client cache

The sections below go into more detail concerning each part of these features.

Windows Update Agent

The Windows Update Agent (WUA) was originally released with Windows Server 2000 Service Pack 1 to provide a standard method for detecting, installing, and reporting patch applicability on Microsoft Windows and other Microsoft products like Microsoft Office, Microsoft Exchange, and so on. System Center 2012 R2 Configuration Manager also uses the WUA to detect which updates need to be installed on a Windows system and installs these updates after Configuration Manager has distributed them to the systems. The idea behind this approach is to eliminate the need for a separate scanning engine like the Inventory Tool for Microsoft Updates (ITMU), which was included in Microsoft Systems Management Server 2003.

The WUA is responsible for scheduling and initializing scan, detection, download, and install of updates on the client machine. WUA consists primarily of the following features:

- WUAUCLT.exe
- WUAServ.dll
- WUAeng.dll

WUAUCLT.exe

This utility is the Automatic Updates client and is responsible for all of the user interactions involved when deploying software updates, for example displaying the UI for download and install progress, showing the reboot notification, and so on. The Automatic Updates client communicates with the Automatic Updates engine through a private application programming interface (API).

As a utility you can run, WUAUCLT.exe includes the following switches:

- **/DetectNow** Forces a detection cycle
- **/ShowSettingsDialog** Opens the UI for "Help Protect your PC: turn on Automatic Updates" which allows you to configure Automatic Updates settings
- **/ResetAuthorization** Resets the cookie and the Software Update Services client ID
- **/ResetEulas** Resets the end user license agreements (EULAs) for updates
- **/DemoUI** Launches the System Tray icon

WUAServ.dll

This binary is a stub that hosts the WUASERV service in a SVCHOST.exe process. It is responsible for dynamically loading and unloading WUAeng.dll during the self-update process described later in the section titled "WUA self-update process."

WUAeng.dll

This is the Automatic Updates engine that performs the core tasks of detection and of downloading and installing updates. It does this by communicating with the update agent through the ISUSInternal interface. The Automatic Updates engine is also responsible for receiving notifications from the update agent, for launching the Automatic Updates client (WUAUCLT.exe), for receiving client notifications, and for implementing the private interfaces so that the client can communicate with the Automatic Updates engine.

The Automatic Updates engine consists of the following features:

- **Update Manager** This caches all the applicable updates synced during a scan. It provides the ability to search the cache for updates that satisfy different criteria, to manage updates, and to save updates to an internal database and retrieve them from the database during service startup.
- **Settings Manager** This stores and retrieves Automatic Updates settings. Automatic Updates settings are stored in the registry.
- **Handlers** These implement an interface through which the client engine can provide notification of status concerning asynchronous operations. Handlers invoke the client engine through the ISUSInternal interface to perform their operations.

WUA self-update process

The WUA itself is updated regularly to ensure it can properly detect the latest software updates released by Microsoft. The process by which WUA auto-updates with a new agent version is known as the WUA self-update process. This functionality only works when the Automatic Updates feature of WUA is enabled and is not applicable when Configuration Manager is integrated with Windows Server Update Services (WSUS), since most administrators disable automatic updates from a Group Policy Object (GPO).

The WUA self-update process works like this:

1. WUAeng.dll starts the self-update process by communicating with the self-update service on the WSUS server and downloading WUIDENT.cab, which contains the WUIDENT.txt file.
2. The WUIDENT.txt file provides information about what minimum version of files are needed and what tree to use for self-update based on the operation system version and hardware.
3. If an update is needed for the client binaries, then either WUSETUP.cab or WSUS3Setup.cab is called, depending on the version already on the client.
4. The self-update process finishes by saving a report in the EventCache folder and logging an event in ReportingEvents.log, both of which are located at %systemroot%\Software Distribution.

WUA verbose logging

For troubleshooting purposes, enable verbose logging for the WUA by following these steps:

1. Open the Registry Editor (regedit.exe).
2. Navigate to the following registry key: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\Trace

3. Create the following registry value:

   Value name: Flags

   Value type: REG_DWORD

   Value data: 00000007

4. Create the following registry value:

   Value name: Level

   Value type: REG_DWORD

   Value data: 00000004

Field experience

Experience with customers shows that when you are working with Configuration Manager integrated with WSUS, you need to disable the Automatic Updates feature of the WUA. You can do this by following the steps described in KB 2476479, which is found at *http://support.microsoft.com/kb/2476479*. The WUA can be updated using a GPO or by targeting a query-based collection with machines that have the old WUA version using the Configuration Manager application/package. For more information, see the section "Update WUA with Configuration Manager" later in this chapter.

It's important to remember that although Configuration Manager uses WUA, WUA also maintains its own functionality. This can lead to situations like an unexpected reboot or even unwanted installation of updates if WUA is not properly configured since WUA can also be configured through a GPO when using WSUS without Configuration Manager.

When working with Configuration Manager, it's not necessary to configure a GPO for WUA. This is because enabling the software update client settings using the Configuration Manager console (one of the first steps you perform when implementing a software update) automatically creates a local GPO for WUA on every system on which the Configuration Manager client has been deployed.

If you choose to create a GPO for WUA, you must configure the Windows Update Server option to point to the active software update point server in the site or location. If there is an existing GPO that was intended to manage standalone WSUS prior to implementing Configuration Manager in your environment, the GPO could override the local GPO created by Configuration Manager, which can cause issues when the software update client tries to communicate with the software update point server.

IMPORTANT **Remember that WUA is an embedded service in a Windows service (SVCHOST.exe) and is named Windows Update on Windows Vista and above but Automatic Updates Service on earlier Windows versions. Do not disable WUA because it will prevent the Configuration Manager software update client functionality from working properly.**

Update WUA with Configuration Manager

Updating WUA is straightforward with Configuration Manager when you take advantage of the Application Management feature. The process involves downloading the WUA binary files and creating the application or package in Configuration Manager. Complete the following steps:

1. Download the latest WUA version from KB 949104 found at *http://support.microsoft.com/kb/949104*. Microsoft continually revises this KB article by updating it with the latest WUA version.
2. Create the application or package with the WUA you downloaded.
3. Create a deployment and target it to the WUA older version collection.

> ***See also*** *For more information on how to update the WUA version with Configuration Manager, see this post from the Configuration Manager Product Team blog: http://blogs.technet.com/b/configmgrteam/archive/2014/07/14/how-to-install-the-windows-update-agent-on-client-computers.aspx.*

Windows Update data store

The WUA instruction folder is stored on the client side in %systemroot%\SoftwareDistribution. This folder stores all of the update evaluation information that is produced when the software update scanning process runs. The Datastore.edb file, or Windows Update data store, located at %systemroot%\SoftwareDistribution\DataStore is a database that stores various information for the registered services of WUA, such as SMS WUS Handler, Update Metadata, Computer Info, Eula Acceptance, List of Update Related Files, and so on. This information is parsed before the client runs a detection cycle to generate a list of already detected updates and related metadata. The Datastore.edb file hosts both the updates needed as well as those that have already been applied for a machine. This mechanism is intended to provide better performance for the WUA during the detection cycle since it has the entire history of all update evaluation.

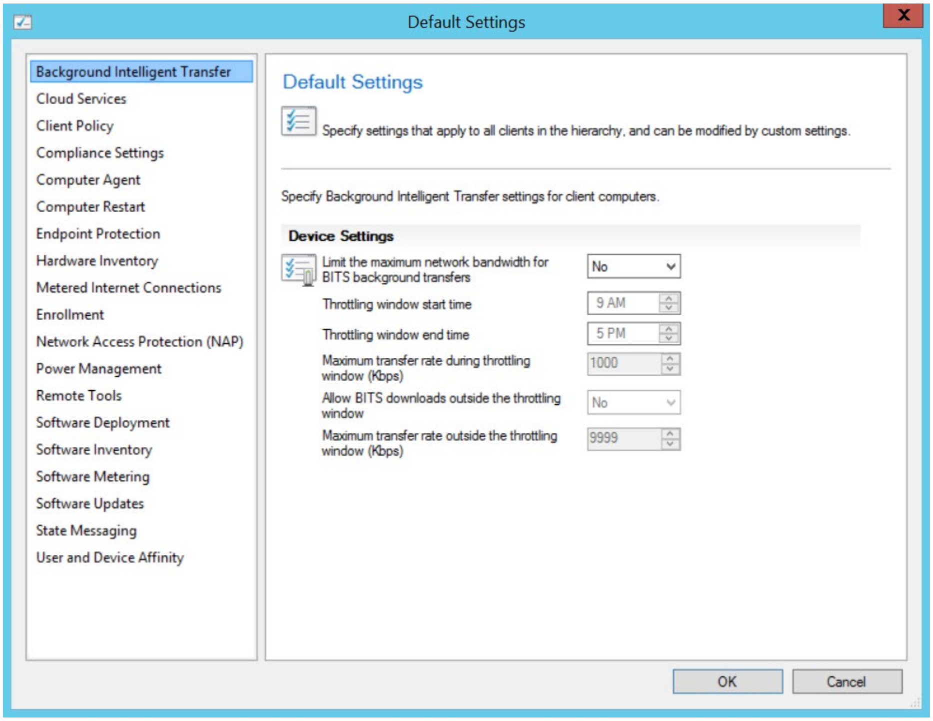
Configuration Manager uses the same mechanism regardless of whether the software update client scan is initiated automatically by the normal schedule or manually by an administrator. A troubleshooting technique frequently used by Microsoft Support is to ask the customer to stop the WUA service and rename or delete the Software Distribution folder. When the WUA service is restarted, the Software Distribution folder is re-created, and the client scanning begins again. This approach provides a fresh start with a new Windows Update data store if the Datastore.edb file is corrupted.

Configuration Manager Software Updates Client Agent

This agent drives the Configuration Manager Software Updates client actions in its Configuration Manager client. The software update client agent configuration can be modified in the Configuration Manager console by navigating to the Administration workspace, then Site Configuration, then Site Settings, and then the Default or Custom client setting. The examples in the sections that follow focus on the settings available in the default settings. In the Default Settings dialog box, a list of settings categories appears on the left and the configurable properties associated with a selected category appear on the left.

Background Intelligent Transfer settings

Since the Configuration Manager client uses Background Intelligent Transfer (BITS) as the default protocol for communicating with the Configuration Manager site server or site system, the first settings to be considered are the BITS throttling settings (located in the Background Intelligent Transfer settings of the Default Settings dialog box shown in Figure 2-1). Consider having BITS throttling settings in place when there are network constraints, such as slow network connections between the Configuration Manager clients and the Configuration Manager software update servers.



Default Settings

Specify settings that apply to all clients in the hierarchy, and can be modified by custom settings.

Specify Background Intelligent Transfer settings for client computers.

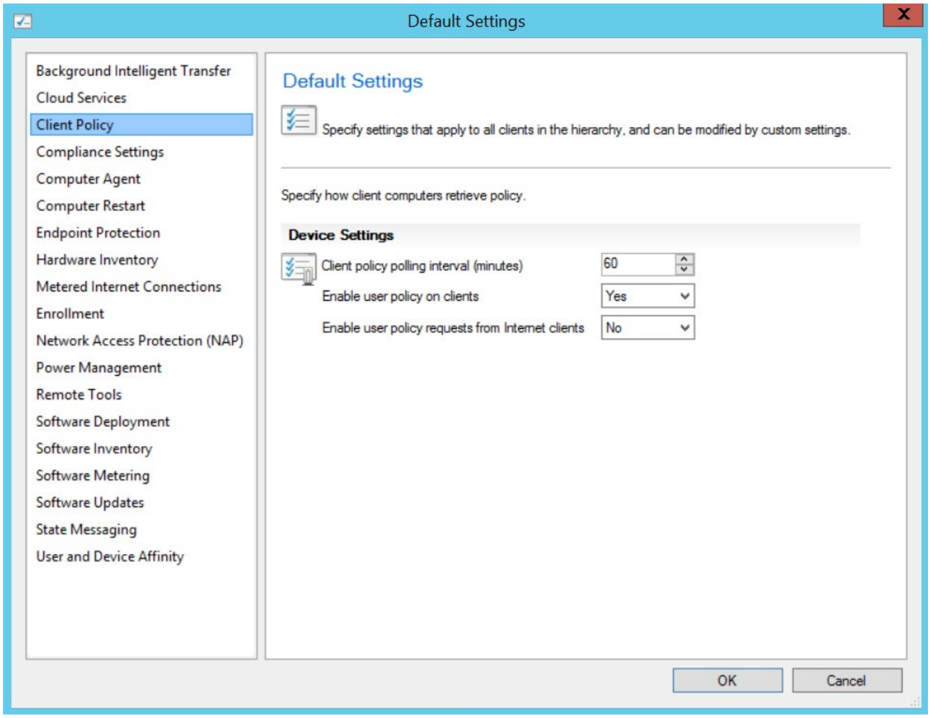
Device Settings

Limit the maximum network bandwidth for BITS background transfers	No
Throttling window start time	9 AM
Throttling window end time	5 PM
Maximum transfer rate during throttling window (Kbps)	1000
Allow BITS downloads outside the throttling window	No
Maximum transfer rate outside the throttling window (Kbps)	9999

FIGURE 2-1 BITS throttling settings in the Configuration Manager console client settings

Client Policy settings

The Client Policy settings, shown in Figure 2-2, govern the frequency with which the Configuration Manager client communicates with the management point to download new software update policies, such as the software update scan schedule or software update deployment.



Default Settings

Background Intelligent Transfer
Cloud Services
Client Policy
Compliance Settings
Computer Agent
Computer Restart
Endpoint Protection
Hardware Inventory
Metered Internet Connections
Enrollment
Network Access Protection (NAP)
Power Management
Remote Tools
Software Deployment
Software Inventory
Software Metering
Software Updates
State Messaging
User and Device Affinity

Default Settings

Specify settings that apply to all clients in the hierarchy, and can be modified by custom settings.

Specify how client computers retrieve policy.

Device Settings

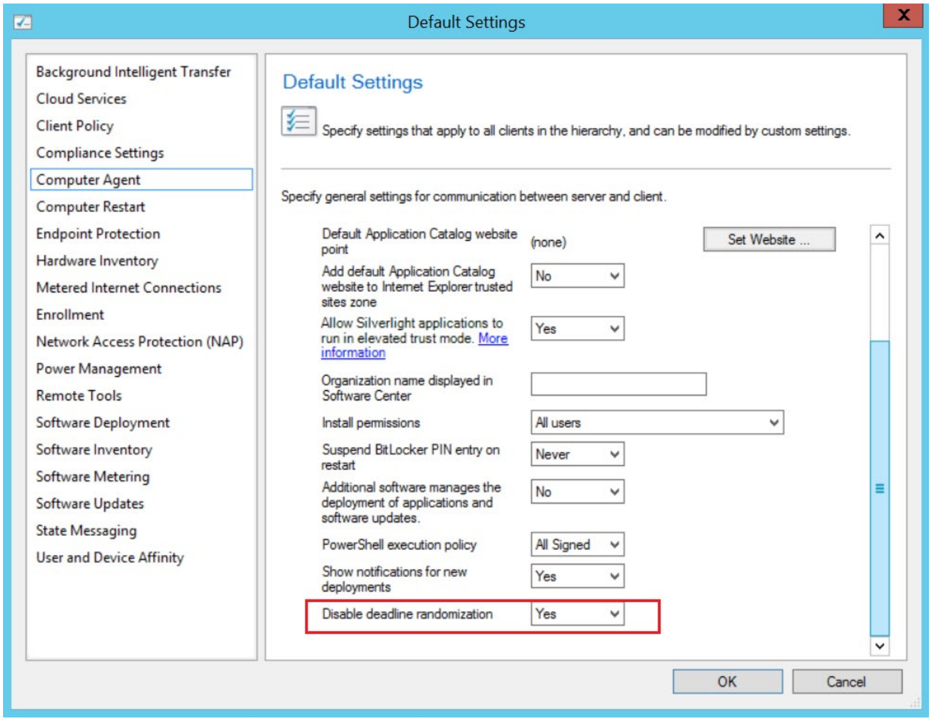
Client policy polling interval (minutes)	60
Enable user policy on clients	Yes
Enable user policy requests from Internet clients	No

FIGURE 2-2 Client Policy settings within the default or custom client settings

Disable Deadline Randomization setting

When you create a software update deployment, you must specify a deadline for that particular deployment. However, the actual time that software update clients start updating installations depends not only on the deadline you specify but also on a setting called Disable Deadline Randomization (located in the Computer Agent workspace of the Default Settings dialog box shown in Figure 2-3).

For example, if you have configured the software update deadline for 02:00 P.M. in your time zone and the Disable Deadline Randomization option is enabled, the software update deployment may begin anytime between 01:00 P.M. and 06:00 P.M. This is because a 4-hour installation randomization is applied to the deadline. This randomization prevents all software update clients from starting update installations at the same time. This can be a particular problem in virtual desktop infrastructure (VDI) environments, but it can also be an issue in a normal virtual environment that has several servers running on a physical host. The Disable Deadline Randomization option is disabled by default in Configuration Manager 2012 R2, but you should enable it for virtual environments, especially VDI environments.



Default Settings

Specify settings that apply to all clients in the hierarchy, and can be modified by custom settings.

Specify general settings for communication between server and client.

Default Application Catalog website point	(none)
Add default Application Catalog website to Internet Explorer trusted sites zone	No
Allow Silverlight applications to run in elevated trust mode. More information	Yes
Organization name displayed in Software Center	
Install permissions	All users
Suspend BitLocker PIN entry on restart	Never
Additional software manages the deployment of applications and software updates.	No
PowerShell execution policy	All Signed
Show notifications for new deployments	Yes
Disable deadline randomization	Yes

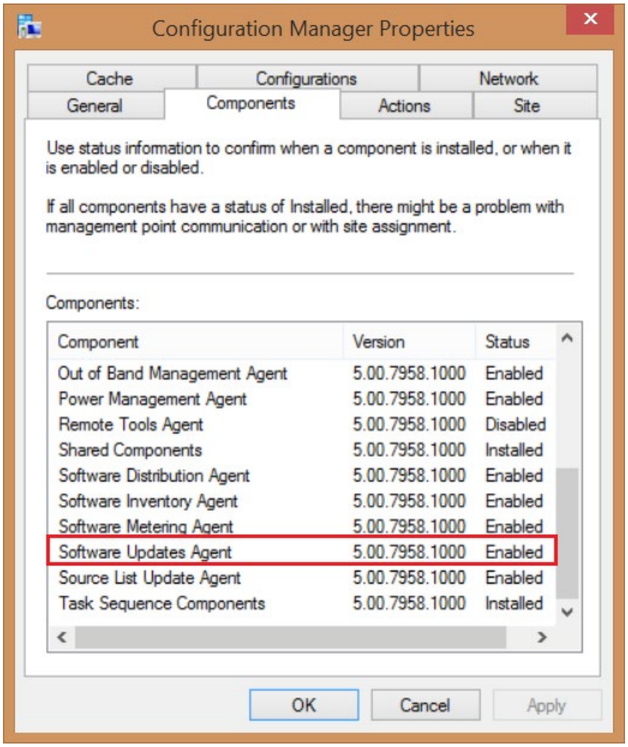
FIGURE 2-3 Computer Agent settings within the default or custom client settings

Enable Software Updates On Clients

Enabling the Enable Software Updates On Clients option (located in the Software Updates workspace of the Default Settings dialog box) causes the following changes:

1. On the site server side, the site control file changes in the SMS_CI_ClientComp class and updates the management point policy in the SQL database. (The site control file is not stored in the file system as of Configuration Manager 2012; it is an XML file stored in the SQL database.)
2. On the client side, when the Configuration Manager client initiates communication with the management point, the client receives the new policy, which includes the software update client feature installation instructions to be installed or applied on the Configuration Manager client.
3. The client then saves the policy results in Windows Management Instrumentation (WMI) and compiles the necessary .MOF files responsible for enabling the software update client feature.

You can verify that the Software Updates Client Agent is enabled by viewing the Components tab of the Configuration Manager Properties client control panel applet shown in Figure 2-4.



Use status information to confirm when a component is installed, or when it is enabled or disabled.

If all components have a status of Installed, there might be a problem with management point communication or with site assignment.

Components:

Component	Version	Status
Out of Band Management Agent	5.00.7958.1000	Enabled
Power Management Agent	5.00.7958.1000	Enabled
Remote Tools Agent	5.00.7958.1000	Disabled
Shared Components	5.00.7958.1000	Installed
Software Distribution Agent	5.00.7958.1000	Enabled
Software Inventory Agent	5.00.7958.1000	Enabled
Software Metering Agent	5.00.7958.1000	Enabled
Software Updates Agent	5.00.7958.1000	Enabled
Source List Update Agent	5.00.7958.1000	Enabled
Task Sequence Components	5.00.7958.1000	Installed

FIGURE 2-4 Configuration Manager client control panel applet with the software update client agent enabled

Software Update Scan Schedule setting

The Software Update Scan Schedule setting (located in the Software Updates workspace of the Default Settings dialog box shown in Figure 2-5) controls the software update scan cycle on the Configuration Manager clients, which occurs every seven days by default but can be changed as necessary to fit your organization's requirements.

Schedule Deployment Re-evaluation setting

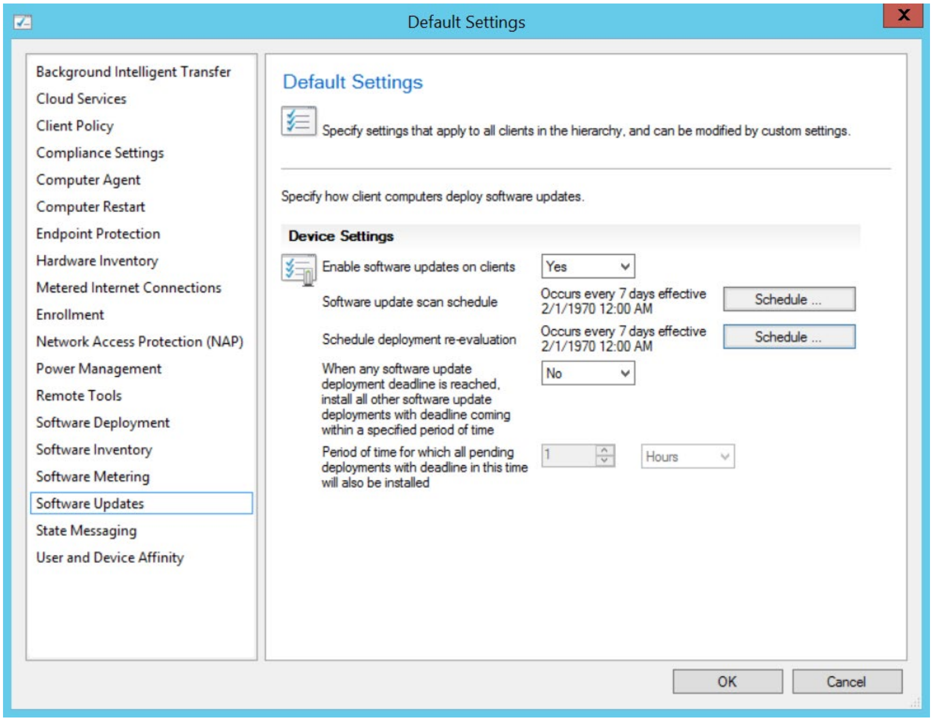
If you previous deployed the software update client throughout your organization and for some reason these deployments were removed from the client machines, the Schedule Deployment Re-evaluation setting (located in the Software Updates workspace of the Default Settings dialog box shown in Figure 2-5) can be used to re-evaluate these previous software update deployments and re-install them for the clients according to a specified schedule.

When Any Software Update Deadline Is Reached, Install All Other Software Update Deployments With Deadline Coming Within A Specified Period Of Time setting

If a number of updates with different deployment deadlines need to be installed for a client but are queued up, you can configure the When Any Software Update… setting (located in the Software Updates workspace of the Default Settings dialog box shown in Figure 2-5) so that after the deadline for the first update on the list is reached, the remaining updates pending installation are installed within their subsequent deadlines.

Period Of Time For Which All Pending Deployments With Deadline In This Time Will Also Be Installed setting

This setting (located in the Software Updates workspace of the Default Settings dialog box shown in Figure 2-5) can be used with the previous one to define a period of time for checking the upcoming deadline for an update.



Default Settings

Specify settings that apply to all clients in the hierarchy, and can be modified by custom settings.

Specify how client computers deploy software updates.

Device Settings

Enable software updates on clients	Yes	
Software update scan schedule	Occurs every 7 days effective 2/1/1970 12:00 AM	Schedule ...
Schedule deployment re-evaluation	Occurs every 7 days effective 2/1/1970 12:00 AM	Schedule ...
When any software update deployment deadline is reached, install all other software update deployments with deadline coming within a specified period of time	No	
Period of time for which all pending deployments with deadline in this time will also be installed	1	Hours

FIGURE 2-5 Software Updates settings example

Windows Management Instrumentation

Windows Management Instrumentation (WMI) has been a native feature in all versions of Microsoft Windows since Windows NT 4.0. WMI acts like a database inside the operating system that can be managed locally or remotely to query operating system information such as hard disk information, the DVD-ROM manufacturer, the applications installed, and so on.

Configuration Manager relies on WMI on both the server and client sides. Any changes that are made in the Configuration Manager console are saved first in WMI, after which the SMS provider commits them to the SQL database. On the client side, this includes all client information, such as the management point, the site code, and other Configuration Manager information, all of which is saved in WMI. All the software update scan evaluation results are also saved in WMI when the client goes through the scanning process, which is described later in this chapter in the section titled "Software update scanning process." All information about software update deployments are also saved in WMI.

Software update policy in WMI

Software update policy is located in two classes in the WMI namespace:

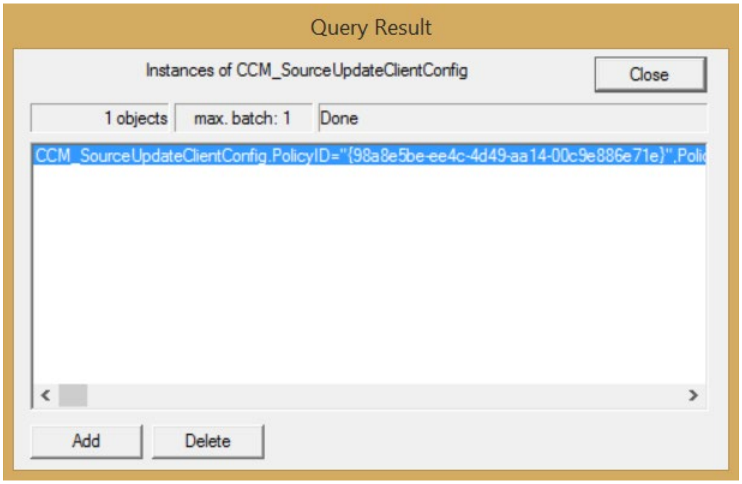
- **Root\ccm\policy\machine\requestedconfig** When the client downloads the policy from the management point, it saves it in this WMI class so the Configuration Manager client Policy Evaluator can evaluate the policy.
- **Root\ccm\policy\machine\actualconfig** Policy Evaluator first checks whether there is a valid policy, and if so, saves it in this class. The actualconfig class is therefore the running configuration for the client, which means it contains all the current settings as defined by the Configuration Manager administrator.

The next two sections show how to use the built-in WBEMTEST.exe utility to view the details of these two software update policies stored in WMI.

Viewing policy stored in the requestedconfig class

To see the software update policy downloaded into the requestedconfig class, complete the following steps:

1. Open WBEMTEST.exe as an administrator, and type **root\ccm\policy\machine\requestedconfig** in the Namespace text box.
2. Click Enum Classes, and select Recursive.
3. Scroll down to review the following classes:
    - CCM_SoftwareUpdatesClientConfig
    - CCM_SourceUpdateClientConfig
    - CCM_UpdateCIAssignment
    - CCM_UpdateSource
4. Double-click a class to reveal details about it. For example, double-click the CCM_SoftwareUpdatesClientConfig class and select Instances to see the details of the software update policy downloaded to the machine, as shown in Figure 2-6.



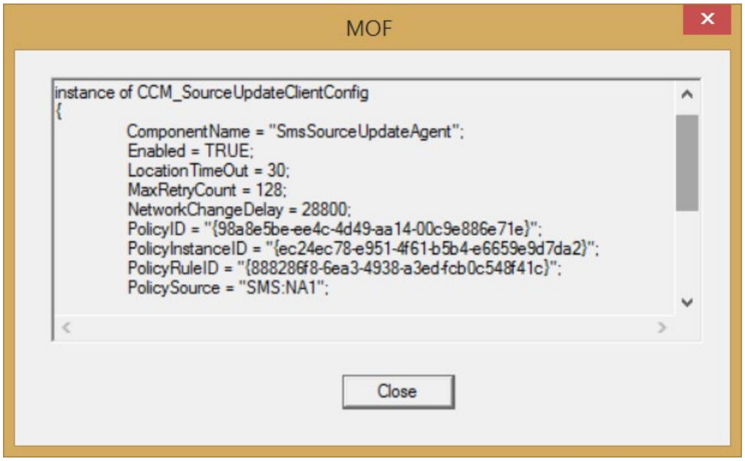
Instances of CCM_SourceUpdateClientConfig

1 objects | max. batch: 1 | Done

```
CCM_SourceUpdateClientConfig.PolicyID="{98a8e5be-ee4c-4d49-aa14-00c9e886e71e}",Polic
```

FIGURE 2-6 Software update client policy in WMI

5. Double-click the policy in the Query Result dialog box, and click Show MOF. This reveals more details about the software update client configuration policy, as shown in Figure 2-7.



```
instance of CCM_SourceUpdateClientConfig
{
        ComponentName = "SmsSourceUpdateAgent";
        Enabled = TRUE;
        LocationTimeOut = 30;
        MaxRetryCount = 128;
        NetworkChangeDelay = 28800;
        PolicyID = "{98a8e5be-ee4c-4d49-aa14-00c9e886e71e}";
        PolicyInstanceID = "{ec24ec78-e951-4f61-b5b4-e6659e9d7da2}";
        PolicyRuleID = "{888286f8-6ea3-4938-a3ed-fcb0c548f41c}";
        PolicySource = "SMS:NA1";
```

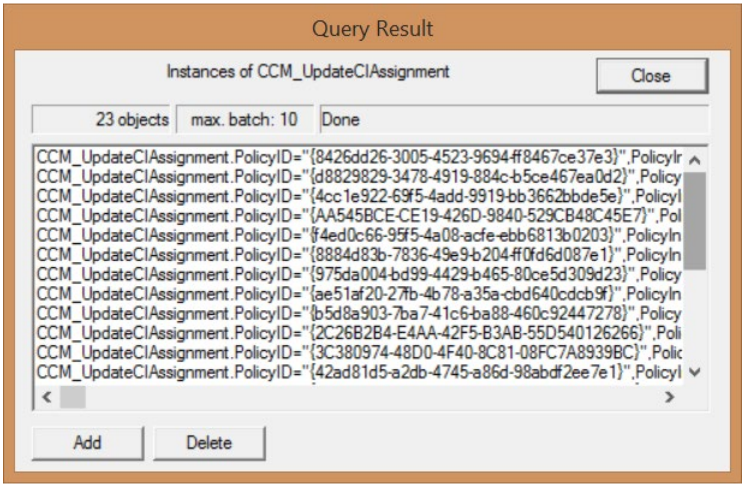
FIGURE 2-7 Software update client policy in WMI

Viewing policy stored in the actualconfig class

To see the software update policy downloaded into the actualconfig class, complete the following steps:

1. Open WBEMTEST.exe as an administrator, and type **root\ccm\policy\machine\actualconfig** in the Namespace text box.
2. Click Enum Classes, and select Recursive.

3. Scroll down to review the following classes:
   - CCM_SoftwareUpdatesClientConfig
   - CCM_SourceUpdateClientConfig
   - CCM_UpdateCIAssignment
   - CCM_UpdateSource
4. Double-click one of the classes and select Instances to reveal all the software update deployments assigned to a machine, as shown in Figure 2-8.



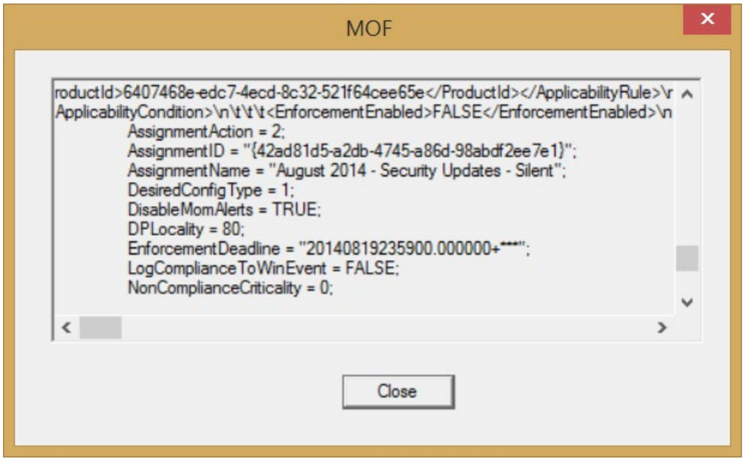
Instances of CCM_UpdateCIAssignment

23 objects | max. batch: 10 | Done

```
CCM_UpdateCIAssignment.PolicyID="{8426dd26-3005-4523-9694-ff8467ce37e3}",PolicyIr
CCM_UpdateCIAssignment.PolicyID="{d8829829-3478-4919-884c-b5ce467ea0d2}",Policy
CCM_UpdateCIAssignment.PolicyID="{4cc1e922-69f5-4add-9919-bb3662bbde5e}",PolicyI
CCM_UpdateCIAssignment.PolicyID="{AA545BCE-CE19-426D-9840-529CB48C45E7}",Pol
CCM_UpdateCIAssignment.PolicyID="{f4ed0c66-95f5-4a08-acfe-ebb6813b0203}",PolicyIn
CCM_UpdateCIAssignment.PolicyID="{8884d83b-7836-49e9-b204-ff0fd6d087e1}",PolicyIn
CCM_UpdateCIAssignment.PolicyID="{975da004-bd99-4429-b465-80ce5d309d23}",Policy
CCM_UpdateCIAssignment.PolicyID="{ae51af20-27fb-4b78-a35a-cbd640cdcb9f}",PolicyIn
CCM_UpdateCIAssignment.PolicyID="{b5d8a903-7ba7-41c6-ba88-460c92447278}",Policy
CCM_UpdateCIAssignment.PolicyID="{2C26B2B4-E4AA-42F5-B3AB-55D540126266}",Poli
CCM_UpdateCIAssignment.PolicyID="{3C380974-48D0-4F40-8C81-08FC7A8939BC}",Polic
CCM_UpdateCIAssignment.PolicyID="{42ad81d5-a2db-4745-a86d-98abdf2ee7e1}",PolicyI
```

FIGURE 2-8 Software update configuration items assigned to a machine in WMI

5. Double-click one of the update assignments in the Query Result dialog box, and select Show MOF to reveal more details about the update assignment, such as assignment name, enforcement deadline, reboot outside maintenance windows, and other useful information, as shown in Figure 2-9.



```
roductId>6407468e-edc7-4ecd-8c32-521f64cee65e</ProductId></ApplicabilityRule>\r
ApplicabilityCondition>\n\t\t\t<EnforcementEnabled>FALSE</EnforcementEnabled>\n
            AssignmentAction = 2;
            AssignmentID = "{42ad81d5-a2db-4745-a86d-98abdf2ee7e1}";
            AssignmentName = "August 2014 - Security Updates - Silent";
            DesiredConfigType = 1;
            DisableMomAlerts = TRUE;
            DPLocality = 80;
            EnforcementDeadline = "20140819235900.000000+***";
            LogComplianceToWinEvent = FALSE;
            NonComplianceCriticality = 0;
```

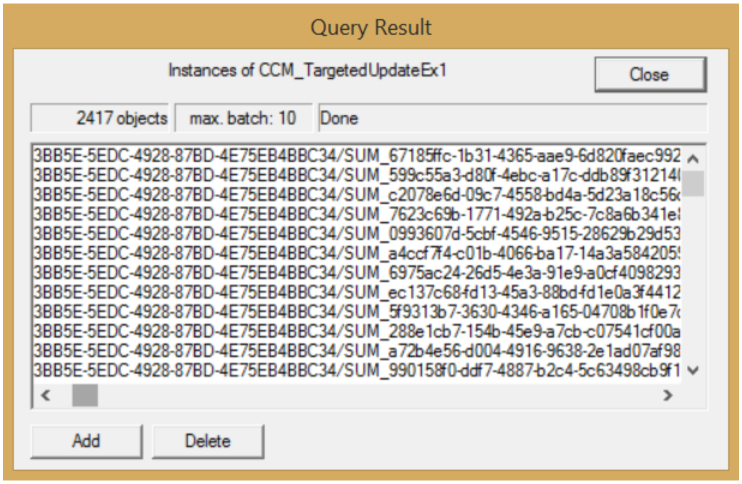
FIGURE 2-9 Software update assignment details

Software update evaluation and deployment in WMI

Evaluation and deployment information concerning software updates is stored both in WMI and in the Windows Update data store, as previously described in this chapter. The WMI information is stored in root\ccm\softwareupdates\deploymentagent and has the following classes:

- **CCM_AssigmentCompliance** After evaluation is finished, the assignment is moved into this class to determine the current compliance state for a particular software update assignment.
- **CCM_AssignmentJobEx1** This class shows an assignment currently being evaluated, which means real-time evaluation only.
- **CCM_DeploymentTaskEx1** This class shows a deployment being installed in real-time on a machine.
- **CCM_TargetedupdateEx1** This class shows every update targeted to a machine after the evaluation process is finished. UpdatesDeployment.log shows the number of updates being targeted to a machine, while this WMI class stores the details about each particular update. Each update has its own unique GUID that can be identified from the Configuration Manager console by adding the Unique Update ID field to the console as described in the section titled “Deployment and update unique ID” later in this chapter.

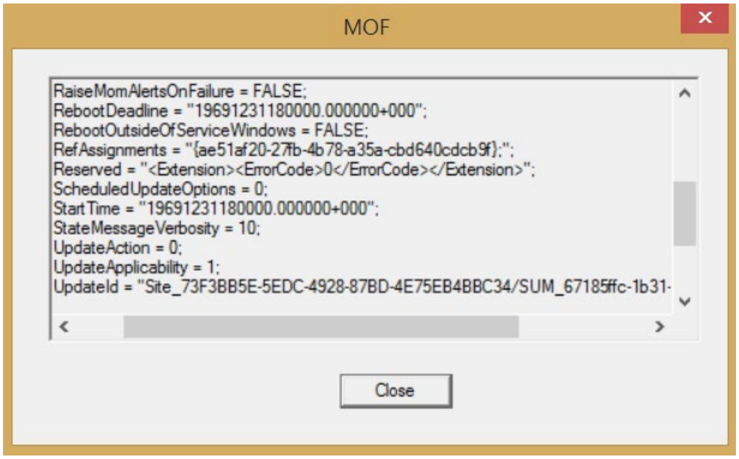
Figure 2-10 shows the results of using WBEMTEST.exe to view the contents of the CCM_TargetedupdateEx1 class. In this example, there are 2,417 updates targeted to the machine.



```
3BB5E-5EDC-4928-87BD-4E75EB4BBC34/SUM_67185ffc-1b31-4365-aae9-6d820faec992
3BB5E-5EDC-4928-87BD-4E75EB4BBC34/SUM_599c55a3-d80f-4ebc-a17c-ddb89f31214
3BB5E-5EDC-4928-87BD-4E75EB4BBC34/SUM_c2078e6d-09c7-4558-bd4a-5d23a18c56
3BB5E-5EDC-4928-87BD-4E75EB4BBC34/SUM_7623c69b-1771-492a-b25c-7c8a6b341e
3BB5E-5EDC-4928-87BD-4E75EB4BBC34/SUM_0993607d-5cbf-4546-9515-28629b29d53
3BB5E-5EDC-4928-87BD-4E75EB4BBC34/SUM_a4ccf7f4-c01b-4066-ba17-14a3a584205
3BB5E-5EDC-4928-87BD-4E75EB4BBC34/SUM_6975ac24-26d5-4e3a-91e9-a0cf4098293
3BB5E-5EDC-4928-87BD-4E75EB4BBC34/SUM_ec137c68-fd13-45a3-88bd-fd1e0a3f4412
3BB5E-5EDC-4928-87BD-4E75EB4BBC34/SUM_5f9313b7-3630-4346-a165-04708b1f0e7
3BB5E-5EDC-4928-87BD-4E75EB4BBC34/SUM_288e1cb7-154b-45e9-a7cb-c07541cf00a
3BB5E-5EDC-4928-87BD-4E75EB4BBC34/SUM_a72b4e56-d004-4916-9638-2e1ad07af98
3BB5E-5EDC-4928-87BD-4E75EB4BBC34/SUM_990158f0-ddf7-4887-b2c4-5c63498cb9f1
```

FIGURE 2-10 Software updates targeted to a machine from WMI

Double-click a specific update to review the details of the update as well as the update unique ID, as shown in Figure 2-11.

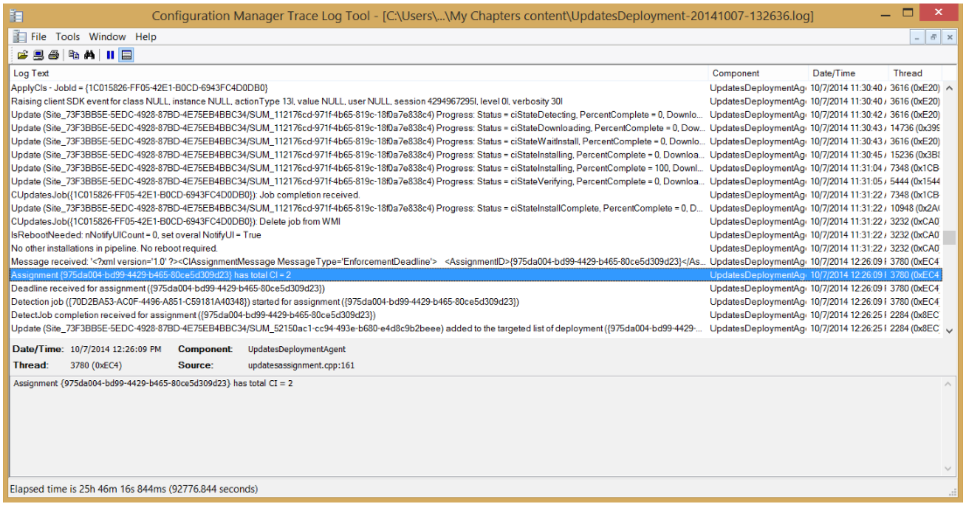


```
RaiseMomAlertsOnFailure = FALSE;
RebootDeadline = "19691231180000.000000+000";
RebootOutsideOfServiceWindows = FALSE;
RefAssignments = "{ae51af20-27fb-4b78-a35a-cbd640cdcb9f};";
Reserved = "<Extension><ErrorCode>0</ErrorCode></Extension>";
ScheduledUpdateOptions = 0;
StartTime = "19691231180000.000000+000";
StateMessageVerbosity = 10;
UpdateAction = 0;
UpdateApplicability = 1;
UpdateId = "Site_73F3BB5E-5EDC-4928-87BD-4E75EB4BBC34/SUM_67185ffc-1b31-
```

FIGURE 2-11 Software update detail targeted to a machine from WMI

Software update logs reference

When troubleshooting software updates from the client side, it's often necessary to cross-reference between WMI and the Configuration Manager client logs. You can use UpdatesDeployment.log to do this since all of the software update deployments and update unique IDs are exposed in this log, as shown in Figure 2-12.


Configuration Manager Trace Log Tool - [C:\Users\...\My Chapters content\UpdatesDeployment-20141007-132636.log]
File
Tools
Window
Help

```
Log Text                                                                                                                                                   Component               Date/Time              Thread
ApplyCIs - JobId = {1C015826-FF05-42E1-B0CD-6943FC4D0DB0}                                                                                                  UpdatesDeploymentAg     10/7/2014 11:30:40 /   3616 (0xE20)
Raising client SDK event for class NULL, instance NULL, actionType 13l, value NULL, user NULL, session 4294967295l, level 0l, verbosity 30l                UpdatesDeploymentAg     10/7/2014 11:30:40 /   3616 (0xE20)
Update (Site_73F3BB5E-5EDC-4928-87BD-4E75EB4BBC34/SUM_112176cd-971f-4b65-819c-18f0a7e838c4) Progress: Status = ciStateDetecting, PercentComplete = 0, Downlo...      UpdatesDeploymentAg     10/7/2014 11:30:42 /   3616 (0xE20)
Update (Site_73F3BB5E-5EDC-4928-87BD-4E75EB4BBC34/SUM_112176cd-971f-4b65-819c-18f0a7e838c4) Progress: Status = ciStateDownloading, PercentComplete = 0, Dow...     UpdatesDeploymentAg     10/7/2014 11:30:43 /   14736 (0x39
Update (Site_73F3BB5E-5EDC-4928-87BD-4E75EB4BBC34/SUM_112176cd-971f-4b65-819c-18f0a7e838c4) Progress: Status = ciStateWaitInstall, PercentComplete = 0, Downlo...   UpdatesDeploymentAg     10/7/2014 11:30:43 /   3616 (0xE20)
Update (Site_73F3BB5E-5EDC-4928-87BD-4E75EB4BBC34/SUM_112176cd-971f-4b65-819c-18f0a7e838c4) Progress: Status = ciStateInstalling, PercentComplete = 0, Downloa...   UpdatesDeploymentAg     10/7/2014 11:30:45 /   15236 (0x3B
Update (Site_73F3BB5E-5EDC-4928-87BD-4E75EB4BBC34/SUM_112176cd-971f-4b65-819c-18f0a7e838c4) Progress: Status = ciStateInstalling, PercentComplete = 100, Downl...   UpdatesDeploymentAg     10/7/2014 11:31:04 /   7348 (0x1CB
Update (Site_73F3BB5E-5EDC-4928-87BD-4E75EB4BBC34/SUM_112176cd-971f-4b65-819c-18f0a7e838c4) Progress: Status = ciStateVerifying, PercentComplete = 0, Downloa...    UpdatesDeploymentAg     10/7/2014 11:31:05 /   5444 (0x1544
CUpdatesJob({1C015826-FF05-42E1-B0CD-6943FC4D0DB0}): Job completion received.                                                                              UpdatesDeploymentAg     10/7/2014 11:31:22 /   7348 (0x1CB
Update (Site_73F3BB5E-5EDC-4928-87BD-4E75EB4BBC34/SUM_112176cd-971f-4b65-819c-18f0a7e838c4) Progress: Status = ciStateInstallComplete, PercentComplete = 0, D...    UpdatesDeploymentAg     10/7/2014 11:31:22 /   10948 (0x2A
CUpdatesJob({1C015826-FF05-42E1-B0CD-6943FC4D0DB0}): Delete job from WMI                                                                                   UpdatesDeploymentAg     10/7/2014 11:31:22 /   3232 (0xCA0
IsRebootNeeded: nNotifyUICount = 0, set overal NotifyUI = True                                                                                             UpdatesDeploymentAg     10/7/2014 11:31:22 /   3232 (0xCA0
No other installations in pipeline. No reboot required.                                                                                                    UpdatesDeploymentAg     10/7/2014 11:31:22 /   3232 (0xCA0
Message received: '<?xml version='1.0' ?><CIAssignmentMessage MessageType='EnforcementDeadline'>   <AssignmentID>{975da004-bd99-4429-b465-80ce5d309d23}</As...  UpdatesDeploymentAg     10/7/2014 12:26:09 I   3780 (0xEC4
Assignment {975da004-bd99-4429-b465-80ce5d309d23} has total CI = 2                                                                                         UpdatesDeploymentAg     10/7/2014 12:26:09 I   3780 (0xEC4
Deadline received for assignment ({975da004-bd99-4429-b465-80ce5d309d23})                                                                                  UpdatesDeploymentAg     10/7/2014 12:26:09 I   3780 (0xEC4
Detection job ({70D2BA53-AC0F-4496-A851-C59181A40348}) started for assignment ({975da004-bd99-4429-b465-80ce5d309d23})                                     UpdatesDeploymentAg     10/7/2014 12:26:09 I   3780 (0xEC4
DetectJob completion received for assignment ({975da004-bd99-4429-b465-80ce5d309d23})                                                                      UpdatesDeploymentAg     10/7/2014 12:26:25 I   2284 (0x8EC
Update (Site_73F3BB5E-5EDC-4928-87BD-4E75EB4BBC34/SUM_52150ac1-cc94-493e-b680-e4d8c9b2beee) added to the targeted list of deployment ({975da004-bd99-4429-...      UpdatesDeploymentAg     10/7/2014 12:26:25 I   2284 (0x8EC

Date/Time:  10/7/2014 12:26:09 PM     Component:  UpdatesDeploymentAgent
Thread:     3780 (0xEC4)              Source:     updatesassignment.cpp:161

Assignment {975da004-bd99-4429-b465-80ce5d309d23} has total CI = 2

Elapsed time is 25h 46m 16s 844ms (92776.844 seconds)
```

FIGURE 2-12 Software updates targeted to a machine shown in UpdatesDeployment.log

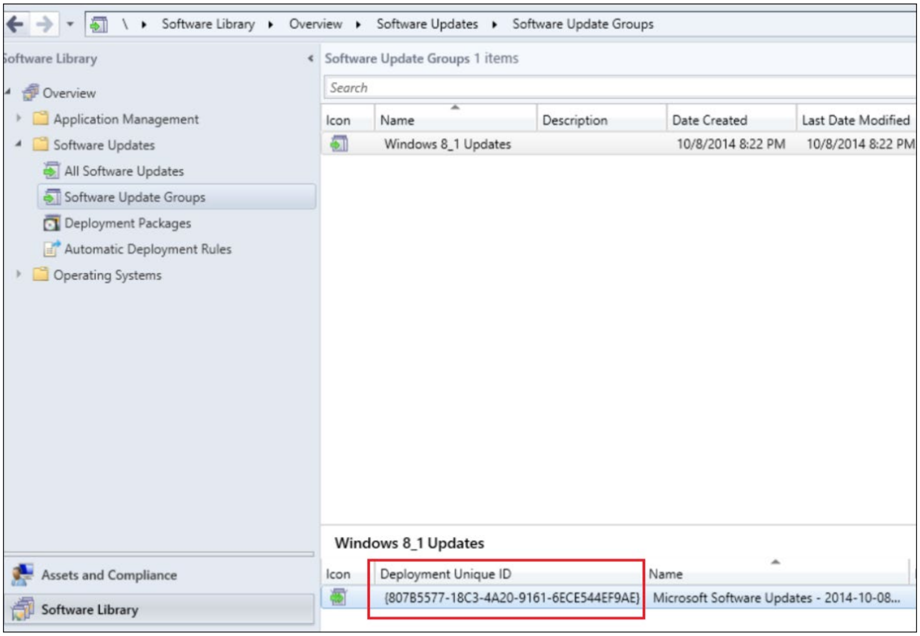
Deployment and update unique ID

You can use the Configuration Manager console to cross-reference software update deployments and update unique IDs. This can be helpful after you have determined which deployment or update is failing on a machine and you need to make sure everything is properly configured on the server side.

To identify the deployment unique ID of a software update, complete the following steps:

1. Open the Configuration Manager console.
2. Navigate to the Monitoring workspace and click Deployments.
3. Right-click on any column and select Deployment ID from the list.

As Figure 2-13 shows, following these steps displays the deployment unique ID that you will need to cross reference with WMI and UpdatesDeployment.log.



Software Update Groups 1 items

Icon	Name	Description	Date Created	Last Date Modified
	Windows 8_1 Updates		10/8/2014 8:22 PM	10/8/2014 8:22 PM

Windows 8_1 Updates

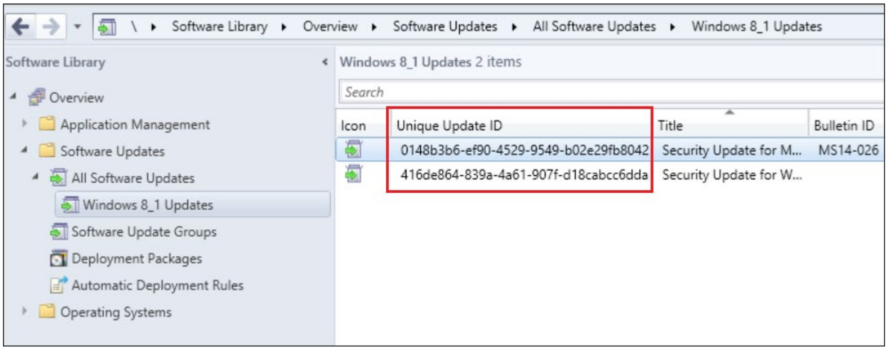
Icon	Deployment Unique ID	Name
	{807B5577-18C3-4A20-9161-6ECE544EF9AE}	Microsoft Software Updates - 2014-10-08...

FIGURE 2-13 The deployment unique ID in the Configuration Manager console

To retrieve the update unique ID, complete the following steps:

1. Open the Configuration Manager console.
2. Navigate to the Software Library workspace, click Software Updates, and then click Software Update Groups.
3. Right-click the Software Update group previously identified with the deployment unique ID and select Show Members.
4. Right-click the column and select Update Unique ID.

As Figure 2-14 shows, following these steps displays the update unique ID that you will need to cross-reference with WMI and UpdatesDeployment.log.



Windows 8_1 Updates 2 items

Icon	Unique Update ID	Title	Bulletin ID
	0148b3b6-ef90-4529-9549-b02e29fb8042	Security Update for M...	MS14-026
	416de864-839a-4a61-907f-d18cabcc6dda	Security Update for W...	

FIGURE 2-14 The update unique ID in the Configuration Manager console

Configuration Manager client cache

By default, the Configuration Manager 2012 R2 client stores deployed packages in the %WinDir%\ccmcache folder, and the default disk space for this folder is 5 gigabytes (GB). Do not encrypt this folder because Configuration Manager cannot upload content to an encrypted folder. The client downloads the software updates from the distribution point using BITS after receiving the policy from the management point.

It's possible to change the Configuration Manager client cache location by specifying the parameter SMSCACHEDIR=C:\WINDOWS\TEMP with CCMSetup.exe during the Configuration Manager client installation. In addition, you can change the cache size by using the parameter SMSCACHESIZE=10240 with CCMSetup.exe, where the value specified is in megabytes.

The content remains in the cache for at least 24 hours. After that time, it can be overwritten by new content if more disk space is necessary. If the Configuration Manager administrator configures a package to persist content in the client cache, the client will not automatically delete or overwrite the package in the cache. In this case, it is necessary to increase the cache size or choose the delete option from the Control Panel applet of the client to delete the content from the cache.

Software update scanning process

Understanding the software update scanning process is important for successfully troubleshooting issues involving the detection and scanning of software updates for deployment to client machines. The software update scanning process includes the following steps:

1. The software update scan is initiated, either manually from the Control Panel applet, by a software update scan cycle action, or based on the Configuration Manager client default or custom settings schedule.
2. The client obtains the WSUS server location from the management point.
3. The Configuration Manager client calls the WUA feature on the client, which then connects to the WSUS server and initiates the compliance scan.
4. The WUA returns a list of the updates that are installed or required on the client. The site server determines what updates are required on each client by comparing the list of all defined updates with the scan results.
5. WSUS stores the results of the scan in the WSUS database.
6. The Configuration Manager client stores the compliance scan results in WMI, and the WUA stores the scan results in Datastore.edb.
7. The Configuration Manager client sends the results to the management point in the form of a batch of state messages.
8. The management point sends the results to the site server where they are then entered in the Configuration Manager site database.

9. The compliance scan data is now available for viewing in the Software Updates Compliance reports and in the Configuration Manager console.

Figure 2-15 shows the numbered steps described in this section.

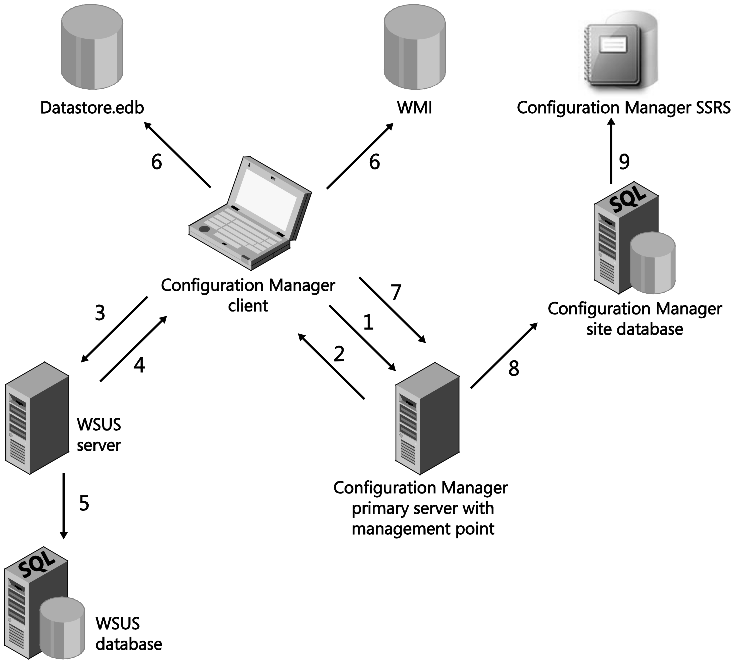

Datastore.edb
WMI
Configuration Manager SSRS
6
6
9
SQL
Configuration Manager client
7
1
3
2
4
Configuration Manager site database
8
WSUS server
5
Configuration Manager primary server with management point
SQL
WSUS database

FIGURE 2-15 An overview of the software update scanning process

Software update installation process

Understanding the software update installation process is important for successfully troubleshooting issues involving installation of software updates on client machines. The software update installation process includes the following steps:

1. A new software update group is created and deployed by the administrator console.
2. The site server requests the software update binaries from the source location as defined in the deployment. This can also be from Microsoft Updates.
3. The site server copies the software update binaries to the package share on the distribution point.
4. The site server adds the new software update deployment to the machine policy and copies the policy to the management point.
5. The client pulls the machine policy from the management point according to the specified schedule and receives the new deployment information. The client then

scans for each software update to verify that they are still required.

6. If a software update is still required, the client requests its binaries from the distribution point. This occurs for each mandatory update, and the binaries obtained are stored in the local cache.
7. The client sends a state message to the management point reporting that the software update was downloaded. The management point then forwards the state message to the site server, which enters the message in the database.

When the software update deadline arrives or the update installation is manually initiated, the client scans for each software update to verify that it is still required.

> **NOTE** **The client scans for and installs the software update using local rules to verify that the update is no longer required. It then sends a state message to the management point to indicate the state of the deployment at completion. For each software update that fails to install, an error status message is sent to the management point. These messages are then forwarded to the site server, which inserts them into the database.**

Figure 2-16 shows the numbered steps described in this section.

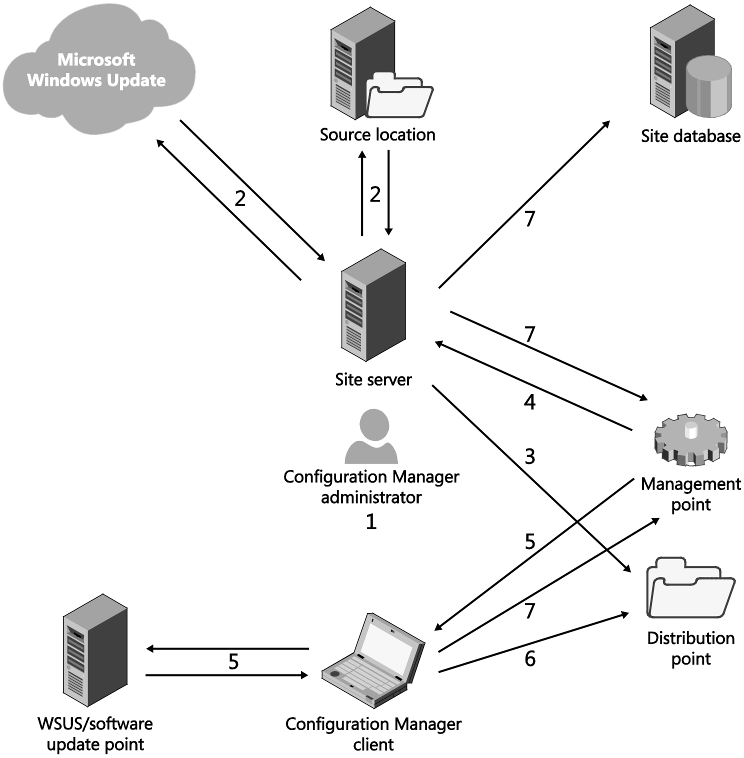


FIGURE 2-16 An overview of the software update installation process

CHAPTER 3

Managing software updates

The task of managing software updates is critical to maintaining the security and operational health of an enterprise and is one of the most important steps an organization can take to secure digital assets. An update management process can help an organization maintain operational effectiveness, mitigate security vulnerabilities, and maintain the integrity of the production environment. Microsoft System Center 2012 R2 Configuration Manager provides a robust vehicle to deliver software updates in a consistent manner.

The patch management process model

Microsoft has developed a four-phased approach to software update management that is designed to give organizations control over the maintenance and deployment of recurrent software update releases. Figure 3-1 illustrates the four phases of the software update management process, which are as follows:

- Phase 1: Assess
- Phase 2: Identify
- Phase 3: Evaluate and Plan
- Phase 4: Deploy

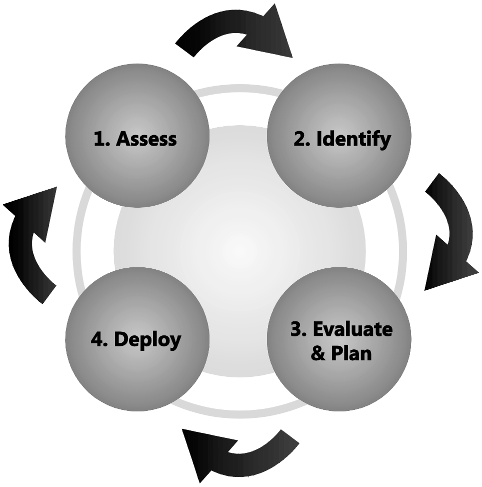


Figure 3-1 The four phases of the software update management process

Phase 1: Assess

The software update management process starts with the Assess phase. This begins with an assessment of the state of the production environment. To effectively patch systems, administrators should strive to understand what security threats and vulnerabilities they are faced with. The assessment is an ongoing process repeated at the beginning of the software update lifecycle to ensure that the environment is continually audited and evaluated.

Typical steps performed during the Assess phase include:

- **Inventory existing computing assets** Differing hardware platforms will likely have different software update requirements, so it is essential to have an up-to-date inventory.
- **Assess security threats and vulnerabilities** The security assessment should include identifying security standards and policies and analyzing system vulnerabilities.
- **Determine the best source for information about new software updates** Sources of information about software updates can include email notifications, alert subscriptions, vendor websites, and security and regulatory organizations.
- **Assess the existing software distribution structure** The assessment should include determining if the software distribution infrastructure in place is sufficient, can be used to distribute software updates, can service the computers in your environment, and is properly maintained.
- **Assess operational effectiveness** Evaluating your operational processes has a direct impact on software update management. Determine whether the operational staff has the training, resources, time, and processes to effectively manage the software update process.

Phase 2: Identify

The purpose of the Identify phase is to discover the applicability of software updates that may be relevant to the operating systems and software in your production environment. This includes prioritizing and identifying the criticality of software updates that may dictate a standard response or initiate an emergency release, as in the case of a zero-day exploit. The trigger for the start of the Identify phase of the software update management process is notification that a new software update exists.

Typical steps performed during the Identify phase include:

- **Identify new software updates** Discovering a new software update starts with notification. Common notification mechanisms include email notifications and security advisory portals.
- **Determine whether software updates are applicable to your production environment** Each software update should be checked for relevance. The first step in checking for relevance is determining whether the software update is designed to

address computers or applications in your production environment. Methods that can be used to determine the applicability of the software update in your environment can include reading security bulletins and Microsoft Knowledge Base (KB) articles, reviewing individual software updates, using the Configuration Manager administrator console, and using Configuration Manager built-in reports.

- **Obtain and verify software update source files** Software update source file verification should include the following steps: verification of the software update source, validation of the software update binaries, review of the accompanying documentation, and verification that the software update is virus free.
- **Determine update priority and submit change request** This step starts the evaluation and planning phase of the software update process for submitting a change request. The change request submission should address the following questions:

  - What is the change?
  - What vulnerability is the change in response to?
  - What services are impacted?
  - Is a restart required?
  - Can the software update be uninstalled?
  - Do countermeasures exist?
  - What are the recommended test strategies?
  - What is the suggested priority?
  - What will be the impact of the change?

  If the change addresses a critical security issue or system instability, the priority of the change request should be marked as "emergency."

Phase 3: Evaluate and Plan

The third phase of the software update management process is the Evaluate and Plan phase. The entry point of this phase is a change request for a software update that has been identified as relevant to your production environment.

Typical steps performed during the Evaluate and Plan phase include:

- **Determine the appropriate response** Prioritize and categorize the change request, and obtain authorization to deploy the software update.
- **Plan the release of the software update** Release planning is the process of determining how to release the software update into the production environment. The major considerations of release planning are determining what needs to be updated, identifying the key issues and constraints, and building the release plan.

- **Build the release** With the release plan in place, the next stage of the phase is to develop scripts, tools, and procedures, which the administrators will use to deploy the software update into the production environment.
- **Conduct acceptance testing of the release** Acceptance testing allows developers and business representatives to check that updates work in an environment that closely mirrors production in that business-critical systems will continue to run successfully after the software update has been deployed. At a minimum, testing should be performed to show that after installation the computer will reboot as designed, the software update can be downloaded and successfully installed, the software update can be successfully uninstalled, and business-critical systems and services continue to run.

Phase 4: Deploy

The fourth phase of the software update management process is the Deploy phase. The entry point of this phase is when a software update package is ready and approvals have been obtained for deployment to the production environment.

Typical steps performed during the Deploy phase include:

- **Deployment preparation** The production environment needs to be prepared for each new release. The steps required for preparing the software update deployment includes communicating a deployment schedule to the organization and assigning and staging distribution points.
- **Deployment of the software updates to targeted computers** The process used to deploy the software update into the production environment depends on the type and nature of the release. Emergency releases typically follow the same process as a normal release but in a compressed timeframe. Ideally, software update deployments utilize a phased rollout, which minimizes the impact of failures or adverse effects introduced by the distribution of a software update.
- **Post-implementation review** The post-implementation review should typically be conducted within one to four weeks of a release deployment to identify improvements that should be made to the update management process. A typical review entails adding vulnerabilities to vulnerability scanning reports and security policy standards, updating build images to include the latest software updates, discussing planned versus actual results, discussing the risks associated with the release, reviewing your organization’s performance through the release, and creating or updating the baseline for your environment.

Understanding software update groups

Software update groups are a new functionality introduced in Configuration Manager 2012 that merge update lists and update deployments into a single object. Software update groups function as containers to which software updates can be added and organized either by product or by timeline. Software updates can be added to software update groups either manually or automatically. The Windows client can assess the applicable Windows updates, so there is no mandatory requirement to separate the software updates by products.

Software update groups provide an effective method to organize software updates in your environment and can be organized to provide baselines for a software update strategy. One approach to using them is to organize your software update groups into the following categories:

- **Reporting group** This compliance group is used for reporting purposes only and is not deployed to the production environment.
- **Rollup group** An initial baseline, this group is used as a starting point to deploy applicable software updates to the production environment.
- **Monthly group** The monthly software update deployment rollout aligns with the Microsoft Patch Tuesday monthly release.
- **Quarterly group** This is a compliance group created once per quarter, containing three months' worth of updates.
- **Yearly group** This compliance group is created at the end of each year to aggregate the previous year's software updates.

Reporting groups

A reporting group is a compliance group, which is a software update group that is not deployed, and it is used to measure all-up compliance. The Configuration Manager console shows the aggregated compliance for all systems. The group can also be used to break down compliance by collection using reports. A compliance group is not limited to 1,000 updates, as is the case of deployed software update groups; however, grouping reporting baselines in some kind of logical grouping is suggested.

Rollup groups

When you begin to deploy software updates, the first task is often to bring your computers to an initial and consistent patch state. A rollup group is used as a starting point for deploying software updates and should include all software updates that are currently applicable. Applicable updates are those that are installed or required on at least one system.

The initial rollup group, once established, is deployed to all production computers capable of being patched. This deployment provides a means to update new computers that do not have the initial baseline patches installed when they are introduced into the environment.

Because this software update group is deployed, it can contain no more than 1,000 updates; therefore, it may be necessary to split this initial baseline into multiple software update groups. In this case, logical groupings should be used. After they are created, you should avoid making frequent changes to these groups to avoid a server-side hit on SQL processing.

Monthly groups

After the initial baseline is been established with the rollup group, monthly software update groups are used to deploy new software updates. Monthly groups are typically aligned with the Microsoft Patch Tuesday release cycle, which is the second Tuesday of each month. The monthly group can be created by using automatic deployment rules or the Distribute Software Updates Wizard.

Automatic deployment rules can be used to create software update groups using an advanced filter expression to identify precisely the updates you want to deploy. The automatic deployment rules can be set to create the monthly deployment in a disabled state so that the administrator can verify the results returned by the automatic process. After it is created, the monthly software update group can be deployed to multiple collections using a phased rollout approach.

There is no need to combine monthly deployment groups into quarterly or yearly deployment groups. Administrators just keep creating monthly updates over time and add the updates from each monthly deployment into quarterly or yearly compliance groups. Avoiding large deployment groups containing hundreds of updates simplifies the task of deploying and troubleshooting software update groups.

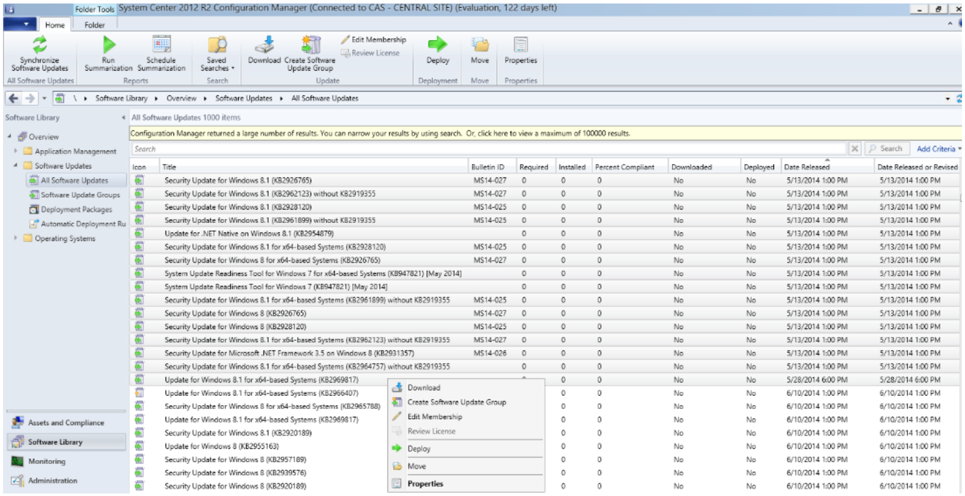
Quarterly and yearly groups

The quarterly and yearly compliance groups are used to provide an aggregate overview of your compliance through the year. This avoids the overhead and deployment summarization caused by adding updates to or modifying deployments in existing deployment groups.

Using software update groups

To create a software update group, complete the following steps:

1. Open the Configuration Manager console, select Software Library, then Overview, then Software Updates, and then All Software Updates. Select multiple software updates. You can use search criteria to find a specific bulletin ID, article ID, or a string that would appear in the title for software updates.
2. In the All Software Updates pane, right-click the selected software updates, and in the context menu, click Create Software Update Group, as shown in the following image.

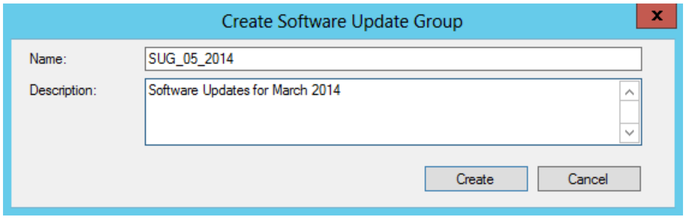


All Software Updates 1000 items

Configuration Manager returned a large number of results. You can narrow your results by using search. Or, click here to view a maximum of 100000 results.

Icon	Title	Bulletin ID	Required	Installed	Percent Compliant	Downloaded	Deployed	Date Released	Date Released or Revised
	Security Update for Windows 8.1 (KB2926765)	MS14-027	0	0	0	No	No	5/13/2014 1:00 PM	5/13/2014 1:00 PM
	Security Update for Windows 8.1 (KB2962123) without KB2919355	MS14-027	0	0	0	No	No	5/13/2014 1:00 PM	5/13/2014 1:00 PM
	Security Update for Windows 8.1 (KB2928120)	MS14-025	0	0	0	No	No	5/13/2014 1:00 PM	5/13/2014 1:00 PM
	Security Update for Windows 8.1 (KB2961899) without KB2919355	MS14-025	0	0	0	No	No	5/13/2014 1:00 PM	5/13/2014 1:00 PM
	Update for .NET Native on Windows 8.1 (KB2954879)		0	0	0	No	No	5/13/2014 1:00 PM	5/13/2014 1:00 PM
	Security Update for Windows 8.1 for x64-based Systems (KB2928120)	MS14-025	0	0	0	No	No	5/13/2014 1:00 PM	5/13/2014 1:00 PM
	Security Update for Windows 8 for x64-based Systems (KB2926765)	MS14-027	0	0	0	No	No	5/13/2014 1:00 PM	5/13/2014 1:00 PM
	System Update Readiness Tool for Windows 7 for x64-based Systems (KB947821) [May 2014]		0	0	0	No	No	5/13/2014 1:00 PM	5/13/2014 1:00 PM
	System Update Readiness Tool for Windows 7 (KB947821) [May 2014]		0	0	0	No	No	5/13/2014 1:00 PM	5/13/2014 1:00 PM
	Security Update for Windows 8.1 for x64-based Systems (KB2961899) without KB2919355	MS14-025	0	0	0	No	No	5/13/2014 1:00 PM	5/13/2014 1:00 PM
	Security Update for Windows 8 (KB2926765)	MS14-027	0	0	0	No	No	5/13/2014 1:00 PM	5/13/2014 1:00 PM
	Security Update for Windows 8 (KB2928120)	MS14-025	0	0	0	No	No	5/13/2014 1:00 PM	5/13/2014 1:00 PM
	Security Update for Windows 8.1 for x64-based Systems (KB2962123) without KB2919355	MS14-027	0	0	0	No	No	5/13/2014 1:00 PM	5/13/2014 1:00 PM
	Security Update for Microsoft .NET Framework 3.5 on Windows 8 (KB2931357)	MS14-026	0	0	0	No	No	5/13/2014 1:00 PM	5/13/2014 1:00 PM
	Security Update for Windows 8.1 for x64-based Systems (KB2964757) without KB2919355		0	0	0	No	No	5/13/2014 1:00 PM	5/13/2014 1:00 PM
	Update for Windows 8.1 for x64-based Systems (KB2969817)		0	0	0	No	No	5/28/2014 6:00 PM	5/28/2014 6:00 PM
	Update for Windows 8.1 for x64-based Systems (KB2966407)			0	0	No	No	6/10/2014 1:00 PM	6/10/2014 1:00 PM
	Security Update for Windows 8 for x64-based Systems (KB2965788)			0	0	No	No	6/10/2014 1:00 PM	6/10/2014 1:00 PM
	Update for Windows 8.1 for x64-based Systems (KB2969817)			0	0	No	No	6/10/2014 1:00 PM	6/10/2014 1:00 PM
	Security Update for Windows 8.1 (KB2920189)			0	0	No	No	6/10/2014 1:00 PM	6/10/2014 1:00 PM
	Update for Windows 8 (KB2955163)			0	0	No	No	6/10/2014 1:00 PM	6/10/2014 1:00 PM
	Security Update for Windows 8 (KB2957189)			0	0	No	No	6/10/2014 1:00 PM	6/10/2014 1:00 PM
	Security Update for Windows 8 (KB2939576)			0	0	No	No	6/10/2014 1:00 PM	6/10/2014 1:00 PM
	Security Update for Windows 8 (KB2920189)			0	0	No	No	6/10/2014 1:00 PM	6/10/2014 1:00 PM

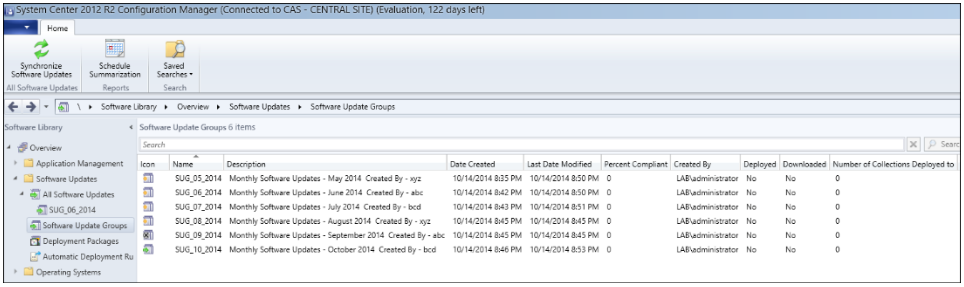
3. In the Create Software Update Group dialog box, enter a name and description for the new software update group as shown in the following image:



Name: SUG_05_2014

Description: Software Updates for March 2014

4. When you click Create in the Create Software Update Group dialog box, you return to the Configuration Manager console. In the console, select Software Library, then Overview, then Software Updates, and then Software Update Groups. In the Software Update Groups pane, you should see the new software update group you created along with any other software update groups that you created previously, as shown in the following imager:



Software Update Groups 6 items

Icon	Name	Description	Date Created	Last Date Modified	Percent Compliant	Created By	Deployed	Downloaded	Number of Collections Deployed to
	SUG_05_2014	Monthly Software Updates - May 2014 Created By - xyz	10/14/2014 8:35 PM	10/14/2014 8:50 PM	0	LAB\administrator	No	No	0
	SUG_06_2014	Monthly Software Updates - June 2014 Created By - abc	10/14/2014 8:42 PM	10/14/2014 8:50 PM	0	LAB\administrator	No	No	0
	SUG_07_2014	Monthly Software Updates - July 2014 Created By - bcd	10/14/2014 8:43 PM	10/14/2014 8:51 PM	0	LAB\administrator	No	No	0
	SUG_08_2014	Monthly Software Updates - August 2014 Created By - xyz	10/14/2014 8:45 PM	10/14/2014 8:45 PM	0	LAB\administrator	No	No	0
	SUG_09_2014	Monthly Software Updates - September 2014 Created By - abc	10/14/2014 8:45 PM	10/14/2014 8:45 PM	0	LAB\administrator	No	No	0
	SUG_10_2014	Monthly Software Updates - October 2014 Created By - bcd	10/14/2014 8:46 PM	10/14/2014 8:53 PM	0	LAB\administrator	No	No	0

Using a phased rollout strategy

Creating a strategy to manage the collections used for software update deployment is an important step in the patch management process. A staged rollout approach provides the best solution to identify problems with software update deployments before they are deployed to the production environment. Here is one strategy your organization might follow:

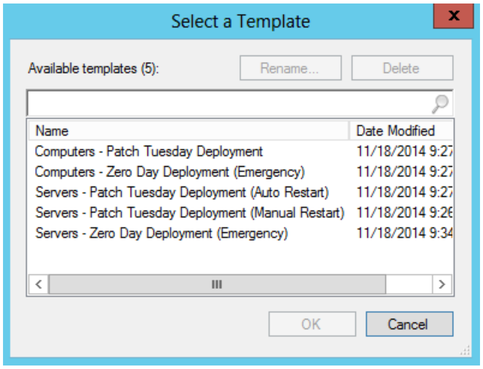
- **Blank for staging** Stage the software update group created with an automatic deployment rule. This collection contains no members.
- **Test workstation** Deploy and test the software update group to a single workstation client.
- **Test server** Deploy and test the software update group to a single server client.
- **Pilot workstations** Deploy and test the software update group to a limited group of workstation clients. It is advantageous for the pilot computers to be representative of the live production environment.
- **Pilot servers** Deploy and test the software update group to a limited group of server clients. Typically, the pilot servers do not host mission-critical applications.
- **Production workstations** Deploy the software update group to production workstations.
- **Production servers** Deploy the software update group to production servers.

Using deployment templates

A deployment template saves time by automatically applying settings and properties to new deployments you create. Since there is no deployment template node in Configuration Manager 2012 R2, it is a good idea to create an empty collection and apply the deployment template to it. When the deployment is created from the template, verify the collection and point the deployment to the appropriate pilot or test collection. Many enterprises frequently create and use templates similar to the ones shown in Figure 3-2. Typical functionality for such templates might be as follows:

- **Computers - Patch Tuesday Deployment** This template is used to deploy the patches released every second Tuesday of the month. This template's settings apply software updates in the maintenance window and provide sufficient time for the user to restart the computer.
- **Computers - Zero Day Deployment (Emergency)** This template is used to immediately deploy a software update. This template includes settings that can install software updates outside of the maintenance window and restart the computer after the software update's installation.

- **Servers – Patch Tuesday Deployment (Auto Restart)** This template is used to deploy updates on the servers as per the normal deployment cycle. With this template, software updates are usually downloaded and installed on the servers automatically.
- **Servers – Patch Tuesday Deployment (Manual Restart)** This template is used to deploy the updates on the servers as per the normal deployment cycle. With this template, software updates are usually downloaded but not installed on the servers automatically. Administrators often prefer to install updates manually and restart the server.
- **Servers – Zero Day Patch Deployment (Emergency)** This template is used to deploy updates on the server when an emergency requires patching to safeguard against a zero-day exploit that has been reported.



Available templates (5):

Name	Date Modified
Computers - Patch Tuesday Deployment	11/18/2014 9:27
Computers - Zero Day Deployment (Emergency)	11/18/2014 9:27
Servers - Patch Tuesday Deployment (Auto Restart)	11/18/2014 9:27
Servers - Patch Tuesday Deployment (Manual Restart)	11/18/2014 9:26
Servers - Zero Day Deployment (Emergency)	11/18/2014 9:34

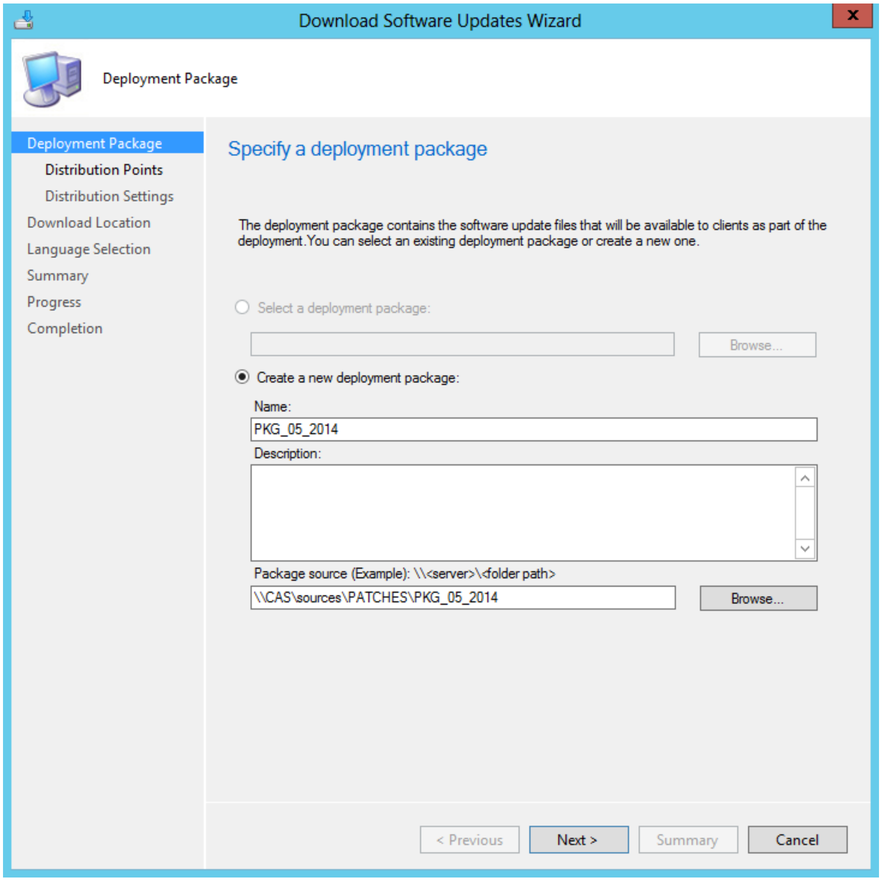
FIGURE 3-2 Examples of patch templates

Using deployment packages

A software update deployment package is used to download software updates from Microsoft to a network shared folder and copy the software update source files to the content library on the site servers and on distribution points that are defined in the deployment.

To create a deployment package from an existing software update group, complete the following steps:

1. Right-click the monthly software update group and click Download to launch the Download Software Updates Wizard.
2. On the Deployment Package page of the wizard, enter the package name and the network path for storing source installers for updates, as shown in the following image:



Specify a deployment package

The deployment package contains the software update files that will be available to clients as part of the deployment.You can select an existing deployment package or create a new one.

○ Select a deployment package:

● Create a new deployment package:

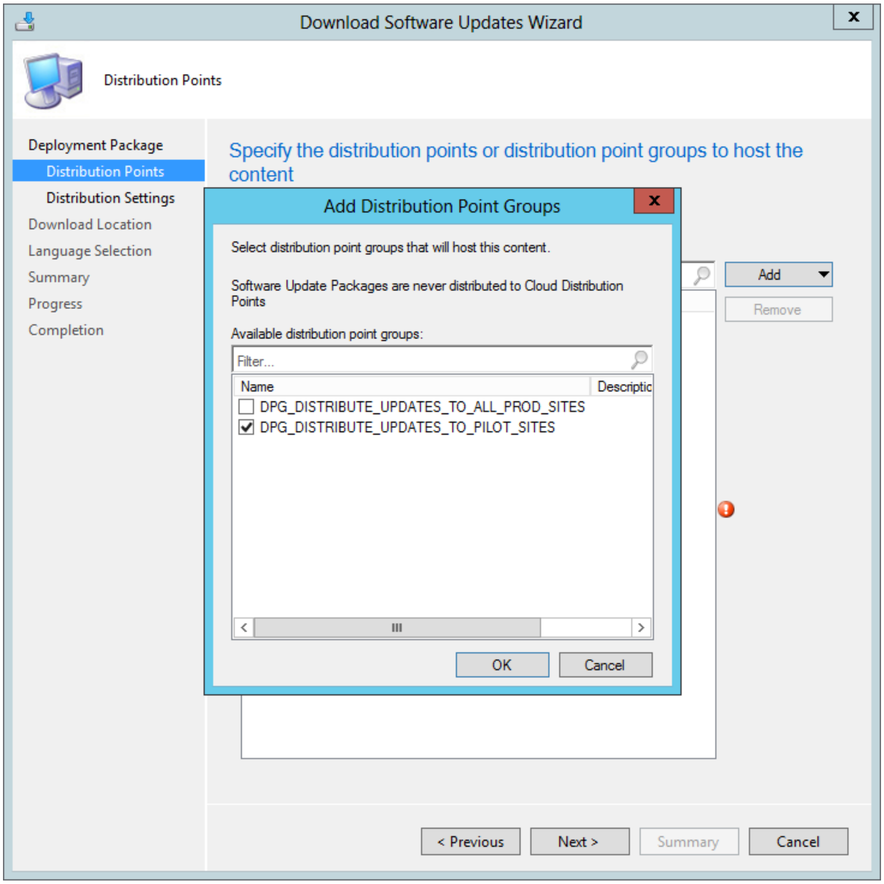
Name: PKG_05_2014

Description:

Package source (Example): \\<server>\<folder path>

\\CAS\sources\PATCHES\PKG_05_2014

3. On the Distribution Points page, click Add to open the Add Distribution Point Groups dialog box. In this dialog box you can select distribution points that will host the content from the list of available distribution points displayed. In the following example, the distribution point where pilot computers are located is selected.



Specify the distribution points or distribution point groups to host the content

Select distribution point groups that will host this content.

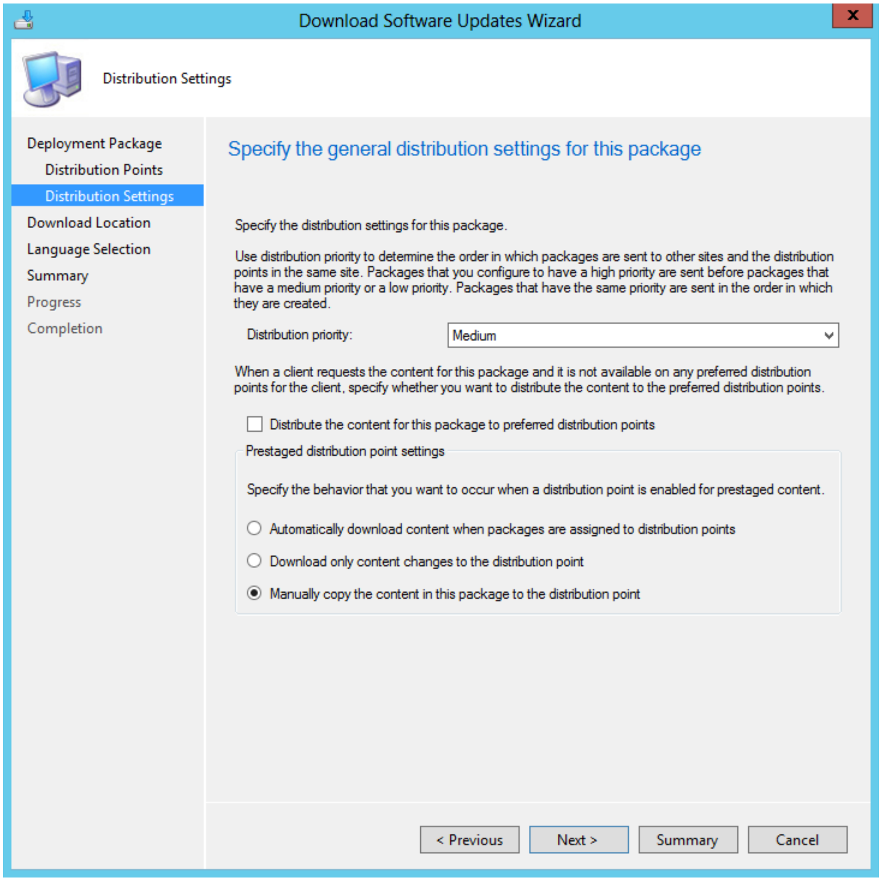
Software Update Packages are never distributed to Cloud Distribution Points

Available distribution point groups:

☐ DPG_DISTRIBUTE_UPDATES_TO_ALL_PROD_SITES
☑ DPG_DISTRIBUTE_UPDATES_TO_PILOT_SITES

TIP It is a good idea to create a distribution point group for pilot locations and all production locations.

4. On the Distribution Settings page, specify the distribution settings for the package according to the Configuration Manager infrastructure design recommendations for your organization:



Specify the general distribution settings for this package

Specify the distribution settings for this package.

Use distribution priority to determine the order in which packages are sent to other sites and the distribution points in the same site. Packages that you configure to have a high priority are sent before packages that have a medium priority or a low priority. Packages that have the same priority are sent in the order in which they are created.

Distribution priority: Medium

When a client requests the content for this package and it is not available on any preferred distribution points for the client, specify whether you want to distribute the content to the preferred distribution points.

□ Distribute the content for this package to preferred distribution points

Prestaged distribution point settings

Specify the behavior that you want to occur when a distribution point is enabled for prestaged content.

○ Automatically download content when packages are assigned to distribution points

○ Download only content changes to the distribution point

● Manually copy the content in this package to the distribution point

5. On the Download Location page, select Download Software Updates From Internet.
6. On the Language Selection page, select the applicable language update files for your organization.
7. On the Summary page, click Next.
8. On the Completion page, verify the details, and then click Close.

Deploying software updates

The two main scenarios for deploying software updates are automatic deployment and manual deployment.

Automatic deployment of software updates

Automatic deployment rules are useful for configuring automatic software updates deployment. Automatic deployment scenarios are used for monthly software updates (Patch

Tuesday). When the rule runs, the software updates that meet a specified criteria are added to a software update group. The source installer files are downloaded and copied to the distribution points. Before deploying to a pilot or production collection, always run on an empty or test collection first. When the test completes successfully, add the pilot or production collection for deployment.

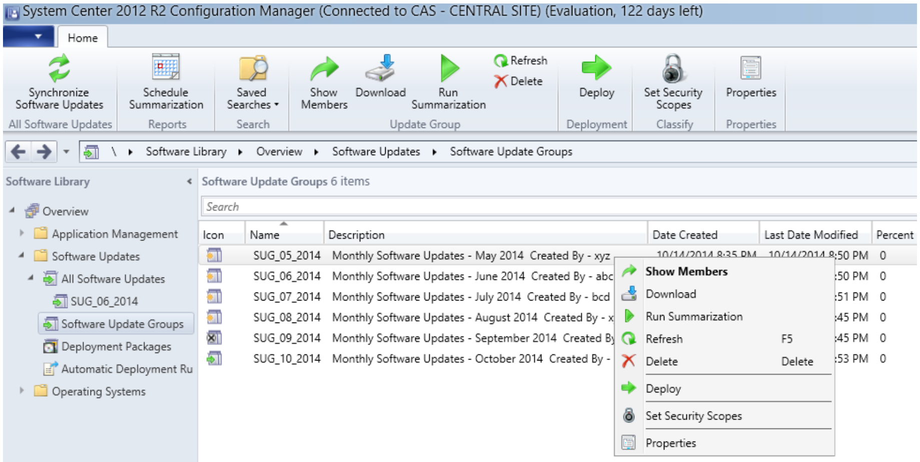
Complete the following steps to automatically deploy the software updates:

1. In the Configuration Manager console, click Software Library, expand Software Updates, and then click Automatic Deployment Rules.
2. Right-click Automatic Deployment Rules and select Create Automatic Deployment Rule to start the Create Automatic Deployment Rule Wizard.
3. On the General page, enter values for Name, Description, and Collection.
4. Select Template – Patch Tuesday. This selection populates the settings automatically.
5. Select Create A New Software Update Group For Patch Tuesday so it will run on monthly basis.
6. Select the Enable The Deployment After The Rule Is Run check box to add the software updates that meet the criteria defined in the rule to a software update group. The content software updates download if necessary. The content is copied to the specified distribution points, and the software updates are deployed to the clients in the target collection.
7. Complete the wizard. Because the Patch Tuesday template was selected, most of the settings are pre-populated, but you can change them as needed to meet your organization’s requirements.

Manual deployment of software updates

To manually deploy software updates using Configuration Manager 2012, complete the following steps:

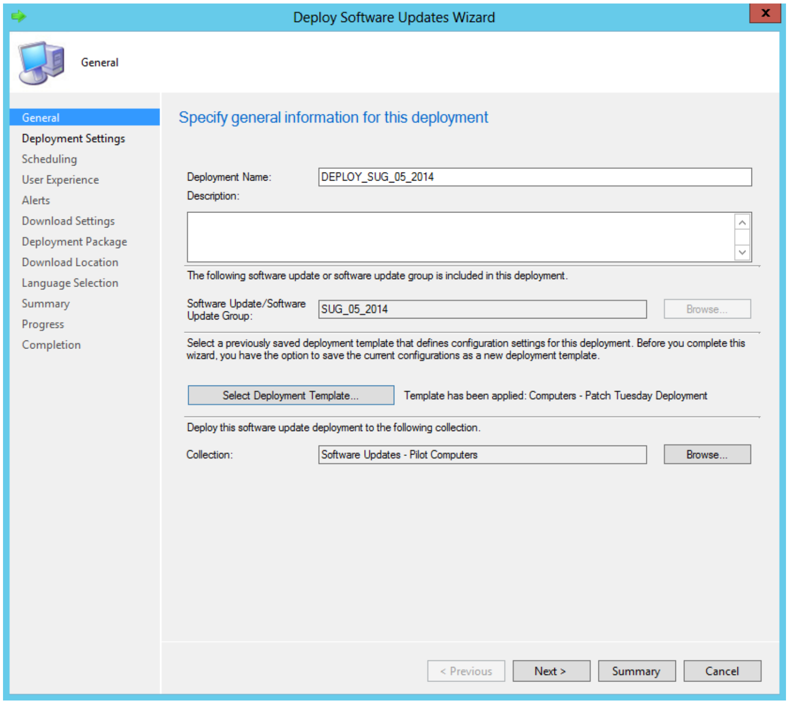
1. In the Configuration Manager 2012 console, select Software Library, then Overview, then Software Updates, and then Software Update Groups.
2. Right-click the software update group you want to use for manually deploying software updates, and from the context menu, select Deploy, as shown in the following image.



Software Update Groups 6 items

Icon	Name	Description	Date Created	Last Date Modified	Percent
	SUG_05_2014	Monthly Software Updates - May 2014 Created By - xyz	10/14/2014 8:35 PM	10/14/2014 8:50 PM	0
	SUG_06_2014	Monthly Software Updates - June 2014 Created By - abc		:50 PM	0
	SUG_07_2014	Monthly Software Updates - July 2014 Created By - bcd		:51 PM	0
	SUG_08_2014	Monthly Software Updates - August 2014 Created By - x		:45 PM	0
	SUG_09_2014	Monthly Software Updates - September 2014 Created By		:45 PM	0
	SUG_10_2014	Monthly Software Updates - October 2014 Created By -		:53 PM	0

3. In the Deploy Software Updates Wizard, on the General page, specify the deployment name and, optionally, a description. Click Select Deployment Template and select a deployment template from the list of available templates displayed in the Select A Template dialog box. Selecting a template automatically populates the required settings. In this walkthrough, a template called Computers - Patch Tuesday Deployment is selected, as shown in the next image:



Specify general information for this deployment

Deployment Name: DEPLOY_SUG_05_2014

Description:

The following software update or software update group is included in this deployment.

Software Update/Software Update Group: SUG_05_2014

Select a previously saved deployment template that defines configuration settings for this deployment. Before you complete this wizard, you have the option to save the current configurations as a new deployment template.

Template has been applied: Computers - Patch Tuesday Deployment

Deploy this software update deployment to the following collection.

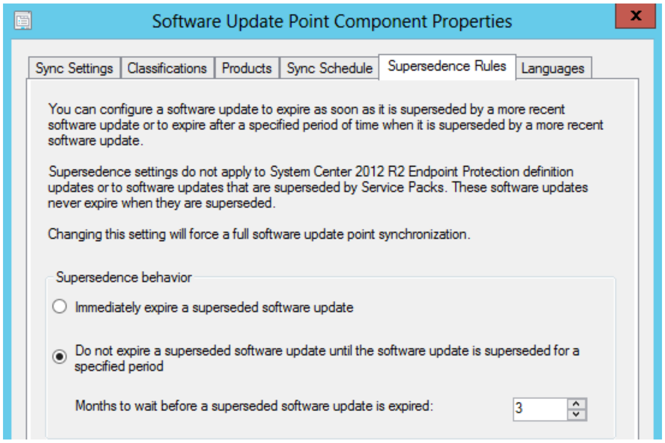
Collection: Software Updates - Pilot Computers

4. On the Deployment Settings page, make sure that Required is selected to create a mandatory software update deployment. After the deployment is created, this option cannot be changed from Required to Available.
5. On the Scheduling page, specify the following:
   - For Software Available Time, select Approximately so that contents are available on all the required distribution points after 24 hours.
   - For Installation Deadline, select 7 days. This adds a random two-hour interval to the scheduled deadline.
6. On the User Experience page, specify the following:
   - For User Notifications, select Display In Software Center, which restricts notifications to only computer restarts.
   - For Deadline Behavior, select the Software Installation and System Restart (if necessary) check boxes only for the Zero Day Patch, which needs to be installed and applied on all the computers immediately.
   - For Device Restart Behavior, if a maintenance window is applied, then there is no need to suppress the reboot for the computers by selecting the Workstations check box. If you have any critical servers that you want to manage manually, then select the Servers check box.
7. On the Alerts page, select the Generate An Alert When The Following Conditions Are Met check box. Set the lower threshold value for client compliance according to the direction from your security team.
8. On the Download Settings page, make selections according to the Configuration Manager 2012 R2 configuration and your available WAN link speed. In case of a zero-day patch, the option to download and install software updates from the distribution point or to download and install software updates from the fallback content source location allows clients to share the content with other clients on the same subnet. Alternatively, you can select the option to download content from Microsoft Update.
9. On the Deployment Package page, select the package that was created when the software updates were downloaded.
10. For Download Location, provide the network path where source installers are stored.
11. On the Language Selection page, select applicable language as per the organization’s requirement.
12. Complete the wizard and verify the messages on the Completion page.

Understanding superseded and expired updates

Software updates expire when they are superseded by more recent software updates or when they are invalidated by Microsoft. System Center 2012 R2 Configuration Manager provides an option to mark superseded software updates as expired immediately or after a selected number of months. Figure 3-3 shows the property settings for configuring supersedence rules, and Figure 3-4 shows an example of a software update that is superseded but not expired.

Although an expired update cannot be deployed, a superseded update can still be installed until it expires.



You can configure a software update to expire as soon as it is superseded by a more recent software update or to expire after a specified period of time when it is superseded by a more recent software update.

Supersedence settings do not apply to System Center 2012 R2 Endpoint Protection definition updates or to software updates that are superseded by Service Packs. These software updates never expire when they are superseded.

Changing this setting will force a full software update point synchronization.

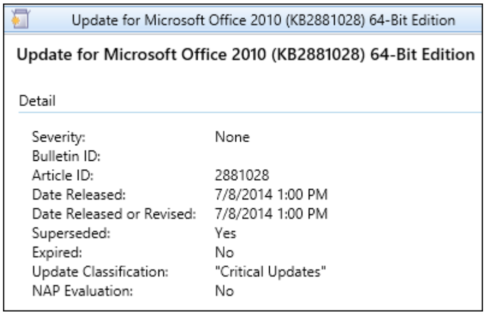
Supersedence behavior

○ Immediately expire a superseded software update

● Do not expire a superseded software update until the software update is superseded for a specified period

Months to wait before a superseded software update is expired: 3

FIGURE 3-3 The Supersedence Rules tab of the Software Update Point Component Properties dialog box



Update for Microsoft Office 2010 (KB2881028) 64-Bit Edition

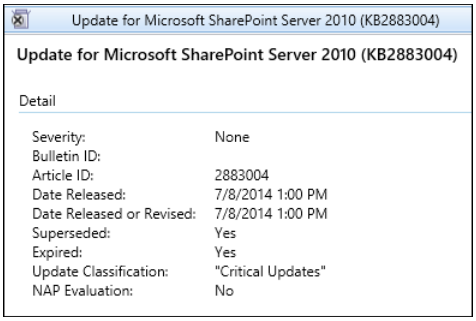
Detail

Severity:	None
Bulletin ID:	
Article ID:	2881028
Date Released:	7/8/2014 1:00 PM
Date Released or Revised:	7/8/2014 1:00 PM
Superseded:	Yes
Expired:	No
Update Classification:	"Critical Updates"
NAP Evaluation:	No

FIGURE 3-4 An example of a superseded but not expired update

An expired update is an update that has been invalidated by Microsoft, for example an update that has caused an issue upon installation. New deployments cannot be created for an expired software update, but existing deployments that contain an expired update continue to work. An expired software update cannot be approved for detection or installation.

Expired updates that are not part of any existing deployments are removed according to the updates cleanup settings. After that, the relevant software update packages are updated automatically. Expired updates that are part of existing deployments are not removed. Figure 3-5 shows an example a software update that is both superseded and expired. Figure 3-6 shows an example of a software update that is superseded but not expired, and Figure 3-7 shows the deletion of expired updates in the wsyncmgr.log.



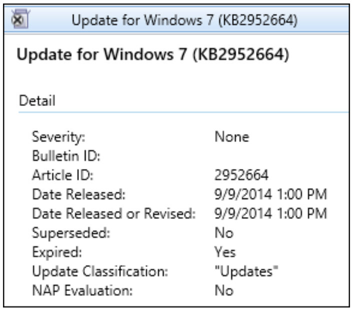
Update for Microsoft SharePoint Server 2010 (KB2883004)

Update for Microsoft SharePoint Server 2010 (KB2883004)

Detail

```
Severity:                  None
Bulletin ID:
Article ID:                2883004
Date Released:             7/8/2014 1:00 PM
Date Released or Revised:  7/8/2014 1:00 PM
Superseded:                Yes
Expired:                   Yes
Update Classification:     "Critical Updates"
NAP Evaluation:            No
```

FIGURE 3-5 Example of a superseded and expired software update



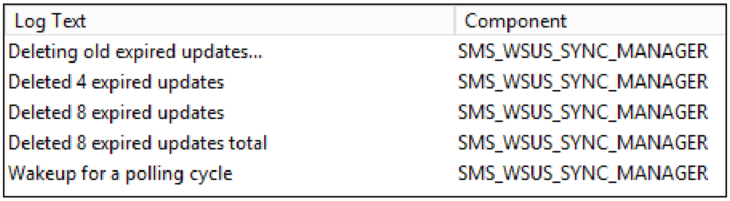
Update for Windows 7 (KB2952664)

Update for Windows 7 (KB2952664)

Detail

```
Severity:                  None
Bulletin ID:
Article ID:                2952664
Date Released:             9/9/2014 1:00 PM
Date Released or Revised:  9/9/2014 1:00 PM
Superseded:                No
Expired:                   Yes
Update Classification:     "Updates"
NAP Evaluation:            No
```

FIGURE 3-6 Example of an expired software update that is not superseded



```
Log Text                              Component
Deleting old expired updates...       SMS_WSUS_SYNC_MANAGER
Deleted 4 expired updates             SMS_WSUS_SYNC_MANAGER
Deleted 8 expired updates             SMS_WSUS_SYNC_MANAGER
Deleted 8 expired updates total       SMS_WSUS_SYNC_MANAGER
Wakeup for a polling cycle            SMS_WSUS_SYNC_MANAGER
```

FIGURE 3-7 Entries in wsyncmgr.log showing the deletion of expired updates

NOTE The behavior of an expired patch cannot be changed. An expired patch should not be installed.

Understanding the expired updates cleanup process

The four phases of removing expired updates and their related content include the expiration action, tomb-stoning, deletion, and source cleanup. At a high level, updates that have been expired and aren't part of an active deployment are deleted seven days after they expire. Configuration Manager 2012 R2 cleans up the source folders automatically.

> ***See also*** *A script to clean up the source folders for versions earlier than Configuration Manager 2012 R2 can be found at http://blogs.technet.com/b/configmgrteam/archive/2012/04/12/software-update-content-cleanup-in-system-center-2012-configuration-manager.aspx.*

Manually removing expired updates

Expired software updates were previously deployable to client computers. Expired updates contained in active deployments continue to be available to clients. When they expire, Configuration Manager does not remove the software updates contained within active software update deployments.

The first step in the process for managing content related to expired updates is getting expired updates out of any deployed update groups. Configuration Manager will never delete any expired update associated with an active deployment.

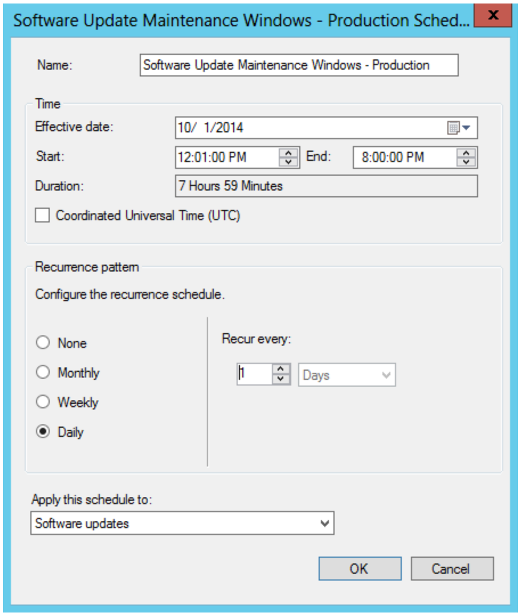
To remove expired updates from deployments, complete the following steps:

1. Navigate to the All Software Updates node under Software Library.
2. To search for all expired updates, add the value for expired updates to search for, leave the default value of Yes, and click Search.
3. The search results include all of your expired updates. Select all of the updates in the list (press CTRL+A), right-click, and then select Edit Membership.
4. The Edit Membership window lists all of the update groups where any of the selected updates from the list are members. To remove the expired updates, clear the selected check boxes.
5. Click OK to remove all of the expired updates from the selected update groups.

Configuring the maintenance window

A maintenance window is a specific timeframe during which various Configuration Manager operations can run on the members of a device collection. Maintenance windows can be applied to all deployments, to software updates, or to task sequences. If a maintenance window is configured for all deployments, then it applies to software updates too, as long as there is no separate software updates maintenance window configured. If a software update maintenance window is configured for a collection, then only this maintenance window is applicable to the client computers in the scope of that collection and all deployment maintenance windows are ignored.

By default, computer restart is blocked outside of an assigned maintenance window, but this can be changed in the deployment settings. Multiple maintenance windows can be assigned to computers, and the start and end times may or may not overlap. Be careful that overlapping maintenance windows do not overlap in such a way that the computers do not come out of maintenance mode. Figure 3-8 shows the settings available for scheduling a maintenance window called Software Update Maintenance Windows - Production. It is a good idea to keep maintenance windows for software updates separate from maintenance windows used for other purposes so that administrators have more control over when software updates deploy and they will not conflict during other deployments.



Software Update Maintenance Windows - Production Sched...

Name: Software Update Maintenance Windows - Production

Time

Effective date: 10/ 1/2014

Start: 12:01:00 PM End: 8:00:00 PM

Duration: 7 Hours 59 Minutes

□ Coordinated Universal Time (UTC)

Recurrence pattern

Configure the recurrence schedule.

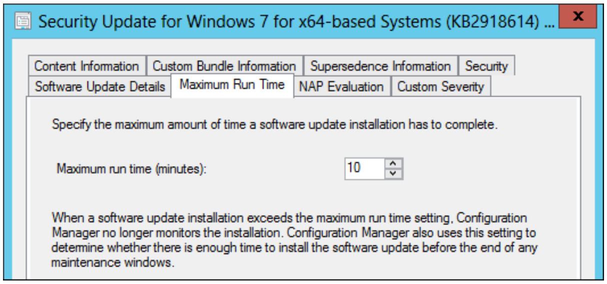
○ None
○ Monthly
○ Weekly
● Daily

Recur every: 1 Days

Apply this schedule to: Software updates

FIGURE 3-8 Software Update Maintenance Windows settings

The maintenance window should be longer than the total run time of all the updates. Check the maximum run time for each software update before you set the maintenance window, as shown in Figure 3-9.



Security Update for Windows 7 for x64-based Systems (KB2918614) ...

Specify the maximum amount of time a software update installation has to complete.

Maximum run time (minutes): 10

When a software update installation exceeds the maximum run time setting, Configuration Manager no longer monitors the installation. Configuration Manager also uses this setting to determine whether there is enough time to install the software update before the end of any maintenance windows.

FIGURE 3-9 Software update maximum run time

This page intentionally left blank

CHAPTER 4

Monitoring software updates

To prove that the Microsoft System Center 2012 R2 Configuration Manager Software Updates feature is doing its part in keeping an organization compliant, you can use monitoring to assure management or upper levels, even the IT auditors, that your patch management is working as expected. Monitoring patch compliance properly has been a pain point for Configuration Manager administrators because the results displayed on the console look different than they do in built-in reports.

This chapter describes the different aspects of monitoring, explains how to use monitoring, and covers different compliance data. The topics that are discussed include the following:

- Compliance accuracy
- Tracking compliance data
- Built-in reports

Compliance accuracy

After software updates are deployed, client systems go through different compliance states. To assess compliance accuracy, administrators require deep understanding of the software update scan workflow, how to track compliance data, and how to customize reports using SQL reporting services. To monitor and track software update deployments, you can use the administrator console, deployment monitoring tools from a System Center 2012 R2 Configuration Manager Toolkit, or reports.

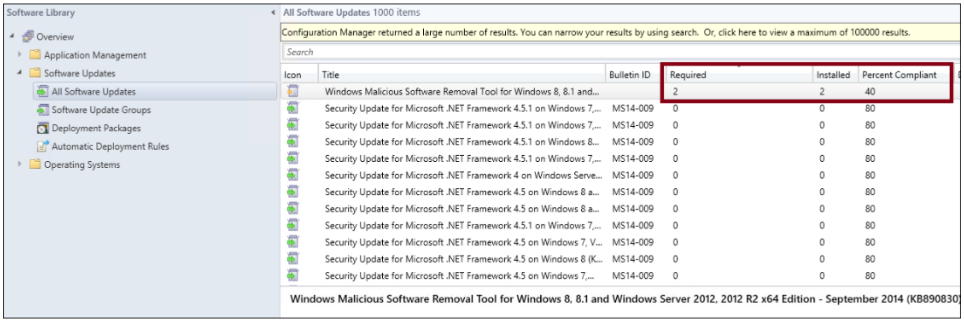
Compliance states from the console

When software updates sync with Microsoft Updates in your Configuration Manager environment, newly added updates in the Configuration Manager console initially show zero values under the Required column. This happens when the client fails to complete the software update scan and send its state messages back to the management point and then to the site server. When the client completes its software update scan and sends its compliance state, state messages are inserted into the site database. The compliance data is then displayed in the Configuration Manager console.

Four types of compliance states can be displayed in the Configuration Manager console:

- **Installed** This means the software update is applicable and the client already has the update installed.
- **Not Required** This means the software update is not applicable to the client system.
- **Required** This means the software update is applicable but is not yet installed. Alternatively, it may mean that the software update was installed but the state message has not yet been uploaded to the site server.
- **Unknown** This means either that the client system did not complete the software scan or the site server did not receive the scan status from the client system.

Being able to properly assess compliance status requires some basic understanding of how compliance percentages are calculated. For example, Figure 4-1 shows All Software Updates selected under the Software Updates node in the administrator console. The selected item shows two required updates, two installed updates, and 40 percent compliance.



All Software Updates 1000 items

Configuration Manager returned a large number of results. You can narrow your results by using search. Or, click here to view a maximum of 100000 results.

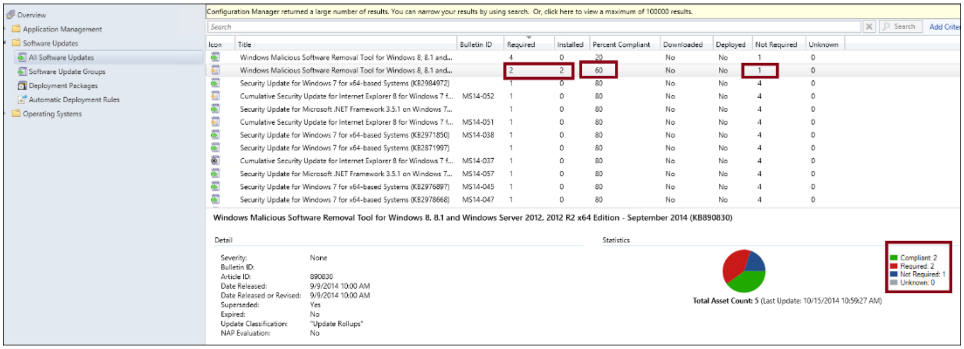
Icon	Title	Bulletin ID	Required	Installed	Percent Compliant
	Windows Malicious Software Removal Tool for Windows 8, 8.1 and...		2	2	40
	Security Update for Microsoft .NET Framework 4.5.1 on Windows 7,...	MS14-009	0	0	80
	Security Update for Microsoft .NET Framework 4.5.1 on Windows 7,...	MS14-009	0	0	80
	Security Update for Microsoft .NET Framework 4.5.1 on Windows 8...	MS14-009	0	0	80
	Security Update for Microsoft .NET Framework 4.5.1 on Windows 7,...	MS14-009	0	0	80
	Security Update for Microsoft .NET Framework 4 on Windows Serve...	MS14-009	0	0	80
	Security Update for Microsoft .NET Framework 4.5 on Windows 8 a...	MS14-009	0	0	80
	Security Update for Microsoft .NET Framework 4.5 on Windows 8 a...	MS14-009	0	0	80
	Security Update for Microsoft .NET Framework 4.5.1 on Windows 7,...	MS14-009	0	0	80
	Security Update for Microsoft .NET Framework 4.5 on Windows 7, V...	MS14-009	0	0	80
	Security Update for Microsoft .NET Framework 4.5 on Windows 8 (K...	MS14-009	0	0	80
	Security Update for Microsoft .NET Framework 4.5 on Windows 7,...	MS14-009	0	0	80

Windows Malicious Software Removal Tool for Windows 8, 8.1 and Windows Server 2012, 2012 R2 x64 Edition - September 2014 (KB890830)

FIGURE 4-1 Software update compliance states

If you do the math, it doesn't seem to add up correctly: if there are two required and two installed updates, how can the item be 40 percent compliant? To understand this, you need to factor in the additional fields Not Required and Unknown. In this case, one system is reporting the status as unknown and zero systems are reporting as not required. If you do the math again, you realize that two out of five are compliant, so 40 percent of the systems are compliant.

If the status of one of the unknown systems changes to not required, as shown in Figure 4-2, two of five systems are compliant, two are still required, and one is not required. Now if you do the math, three out of five are complaint, so 60 percent of the systems are compliant. Not required status is calculated as part of the compliant systems.



Configuration Manager returned a large number of results. You can narrow your results by using search. Or, click here to view a maximum of 100000 results.

Icon	Title	Bulletin ID	Required	Installed	Percent Compliant	Downloaded	Deployed	Not Required	Unknown
	Windows Malicious Software Removal Tool for Windows 8, 8.1 and...		4	0	20	No	No	1	0
	Windows Malicious Software Removal Tool for Windows 8, 8.1 and...		2	2	60	No	No	1	0
	Security Update for Windows 7 for x64-based Systems (KB2984972)		1	0	80	No	No	4	0
	Cumulative Security Update for Internet Explorer 8 for Windows 7 f...	MS14-052	1	0	80	No	No	4	0
	Security Update for Microsoft .NET Framework 3.5.1 on Windows 7...		1	0	80	No	No	4	0
	Cumulative Security Update for Internet Explorer 8 for Windows 7 f...	MS14-051	1	0	80	No	No	4	0
	Security Update for Windows 7 for x64-based Systems (KB2971850)	MS14-038	1	0	80	No	No	4	0
	Security Update for Windows 7 for x64-based Systems (KB2871997)		1	0	80	No	No	4	0
	Cumulative Security Update for Internet Explorer 8 for Windows 7 f...	MS14-037	1	0	80	No	No	4	0
	Security Update for Microsoft .NET Framework 3.5.1 on Windows 7...	MS14-057	1	0	80	No	No	4	0
	Security Update for Windows 7 for x64-based Systems (KB2976897)	MS14-045	1	0	80	No	No	4	0
	Security Update for Windows 7 for x64-based Systems (KB2978668)	MS14-047	1	0	80	No	No	4	0

Windows Malicious Software Removal Tool for Windows 8, 8.1 and Windows Server 2012, 2012 R2 x64 Edition - September 2014 (KB890830)

Detail

Severity:	None
Bulletin ID:	
Article ID:	890830
Date Released:	9/9/2014 10:00 AM
Date Released or Revised:	9/9/2014 10:00 AM
Superseded:	Yes
Expired:	No
Update Classification:	"Update Rollups"
NAP Evaluation:	No

Statistics

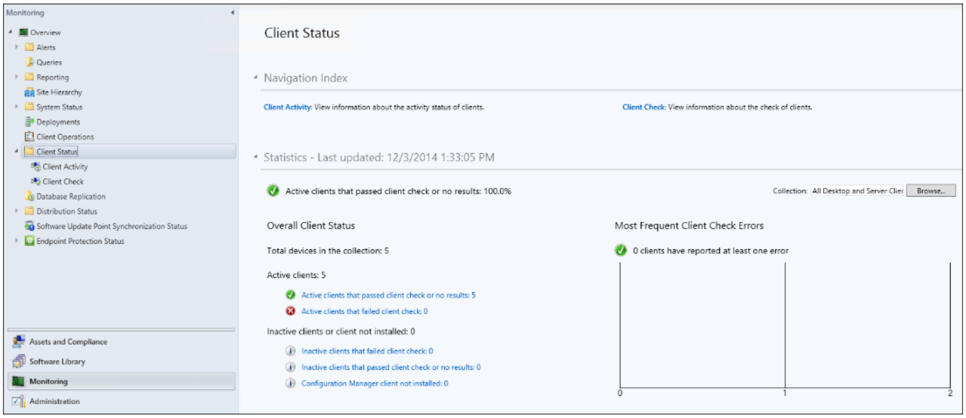
🟩 Compliant: 2
🟥 Required: 2
🟦 Not Required: 1
⬜ Unknown: 0

Total Asset Count: 5 (Last Update: 10/15/2014 10:59:27 AM)

FIGURE 4-2 Software update compliance states with additional Not Required and Unknown columns

Managing client health

Client Status is a built-in feature in the Configuration Manager console. It provides robust information about the state of each service. Before you can take advantage of the Client Status feature, you must configure your Configuration Manager site to specify appropriate parameters that are used to monitor and automatically remediate Configuration Manager clients. You can also configure alerts if client activity falls below a specified threshold. To configure and monitor Client Status in the Configuration Manager console, in the Monitoring workspace, click Client Status. The overall client status details with the number of active clients that passed client check and number of active clients that failed client check displays in the right side of the console. Inactive clients or computers with no Configuration Manager client installed are also indicated, as shown in Figure 4-3.



Client Status

Navigation Index

Client Activity: View information about the activity status of clients.

Client Check: View information about the check of clients.

Statistics - Last updated: 12/3/2014 1:33:05 PM

✓ Active clients that passed client check or no results: 100.0%

Collection: All Desktop and Server Clie

Overall Client Status

Total devices in the collection: 5

Active clients: 5

✓ Active clients that passed client check or no results: 5

☒ Active clients that failed client check: 0

Inactive clients or client not installed: 0

ⓘ Inactive clients that failed client check: 0

ⓘ Inactive clients that passed client check or no results: 0

ⓘ Configuration Manager client not installed: 0

Most Frequent Client Check Errors

✓ 0 clients have reported at least one error

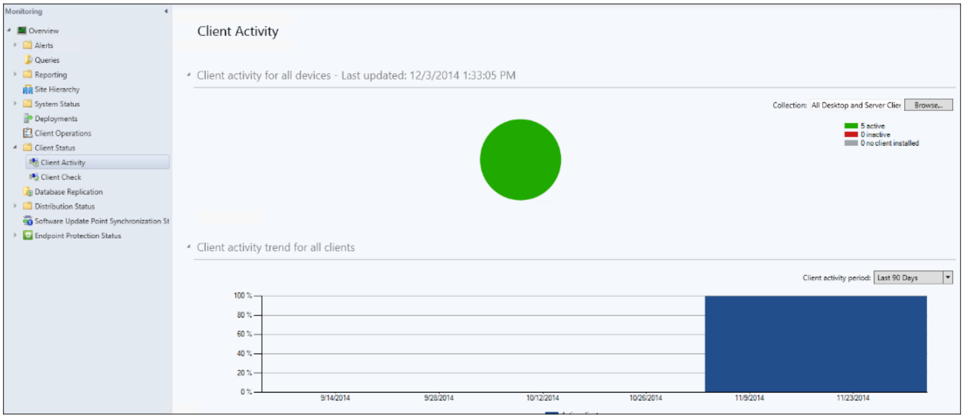
FIGURE 4-3 Overall Client Status view

There are two configuration settings for Client Status: Client Activity and Client Check. Client Activity displays active and inactive systems and computers with no Configuration Manager client installed. To configure the Client Activity setting, complete the following steps:

1. In the Configuration Manager console, navigate to the Monitoring workspace.

2. In the Monitoring workspace, right-click Client Status and select Client Status Settings.
3. Under Client Status Settings Properties, configure different client status checks and their duration.

If none of the criteria are met, the client will be marked as inactive. These settings fall under Client Activity. You can monitor Client Activity as shown in Figure 4-4 by expanding the Client Status node and selecting the Client Activity node in the left pane of the console. The Client Activity node provides overall client activity for all systems located in the All Desktop and Server Clients collection.



Client Activity

Client activity for all devices - Last updated: 12/3/2014 1:33:05 PM

Collection: All Desktop and Server Clier

Client activity trend for all clients

Client activity period: Last 90 Days

FIGURE 4-4 Client Activity node

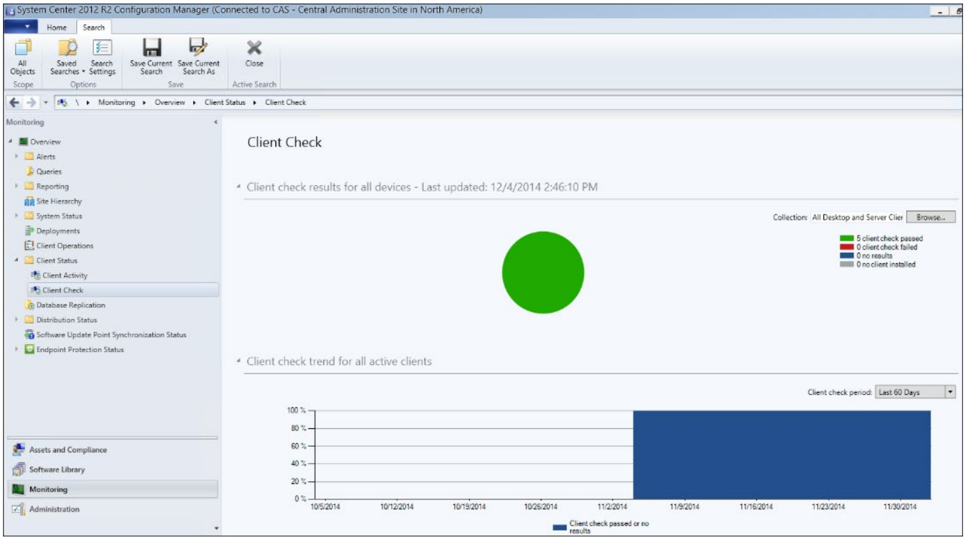
Client Check displays clients that passed or failed the evaluation check. To configure the settings for Client Check, complete the following steps:

1. In the Configuration Manager console, go to the Monitoring workspace.
2. In the Monitoring workspace, expand the Client Status folder, and click Client Check.

In the Client Check node, you can monitor clients that passed the check, clients that failed the check, and clients with no results, as shown in Figure 4-5. This evaluation check is based on the following client evaluation rules, which run as part of the task scheduler every day at midnight by default. If any of the checks fail, then Client Evaluation (CCMEval.exe) automatically tries to remediate and fix the issue.

- Verify/remediate WMI service startup and status
- WMI repository read/write test
- Verify/remediate client WMI provider
- WMI repository integrity test
- WMI event sink test
- Microsoft policy platform WMI integrity test
- Verify/remediate BITS startup type

- Verify/remediate client prerequisites
- Verify/remediate client installation
- Verify/remediate SMS agent host service startup and status
- Verify/remediate Microsoft Policy Platform service startup and status
- Verify/remediate antimalware service startup and status
- Verify/remediate Windows Update service startup type
- Verify/remediate Configuration Manager Remote Control service startup and status
- Verify/remediate Configuration Manager Proxy service startup and status
- Verify/remediate SQL CE database is healthy



System Center 2012 R2 Configuration Manager (Connected to CAS - Central Administration Site in North America)

Client Check

Client check results for all devices - Last updated: 12/4/2014 2:46:10 PM

Collection: All Desktop and Server Clier
Browse...

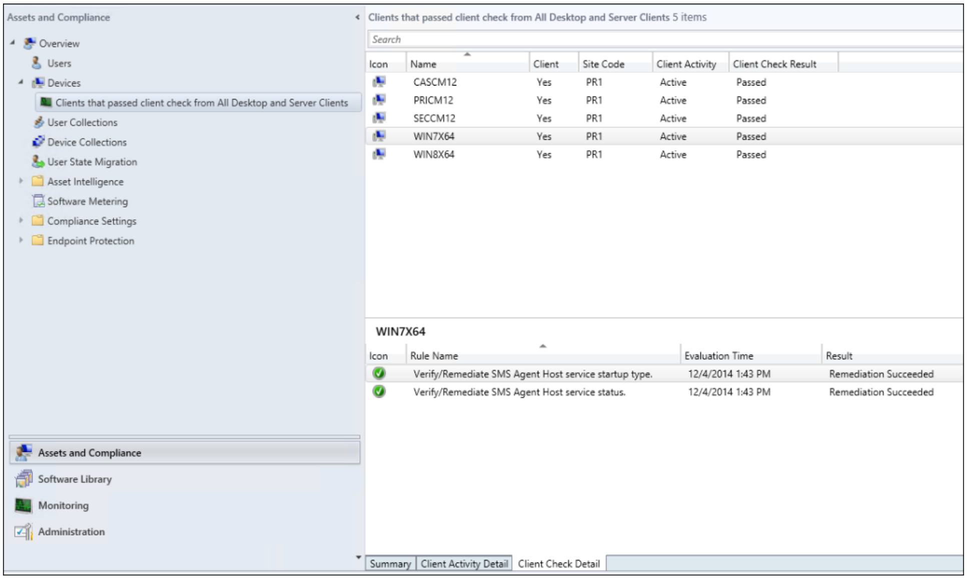
🟩 5 client check passed
🟥 0 client check failed
🟦 0 no results
⬜ 0 no client installed

Client check trend for all active clients

Client check period: Last 60 Days

FIGURE 4-5 Client Check node

You can view the rule that a client ran and its result by clicking on the number of clients that passed or failed the check (for example, 5 Client Check Passed), which is located under Collection in Figure 4-5. When you click this link, another temporary node opens on the left side of the console under the Assets And Compliance workspace. Expand Devices, and then click the Clients That Passed Client Check From All Desktop And Server Clients node, as shown in Figure 4-6. On the right side of the console, select the client, and at the bottom of the workspace, click the Client Check Detail tab to view the rule name and result, as shown in Figure 4-6.



Clients that passed client check from All Desktop and Server Clients 5 items

Icon	Name	Client	Site Code	Client Activity	Client Check Result
	CASCM12	Yes	PR1	Active	Passed
	PRICM12	Yes	PR1	Active	Passed
	SECCM12	Yes	PR1	Active	Passed
	WIN7X64	Yes	PR1	Active	Passed
	WIN8X64	Yes	PR1	Active	Passed

WIN7X64

Icon	Rule Name	Evaluation Time	Result
✓	Verify/Remediate SMS Agent Host service startup type.	12/4/2014 1:43 PM	Remediation Succeeded
✓	Verify/Remediate SMS Agent Host service status.	12/4/2014 1:43 PM	Remediation Succeeded

FIGURE 4-6 Client Check Detail tab

Tracking compliance data

The Configuration Manager Software Updates feature uses state messages to track compliance data. State messaging is new in Configuration Manger and provides client status for software updates, compliance settings, and network access protection. Additionally, administrators can use status messages to track the flow of data through Configuration Manager. State messages are very small and provide specific information regarding the condition of a Configuration Manager client. Understanding state messaging is important since critical software update data relies on it. Figure 4-7 describes the workflow of state messaging for software update compliance data.

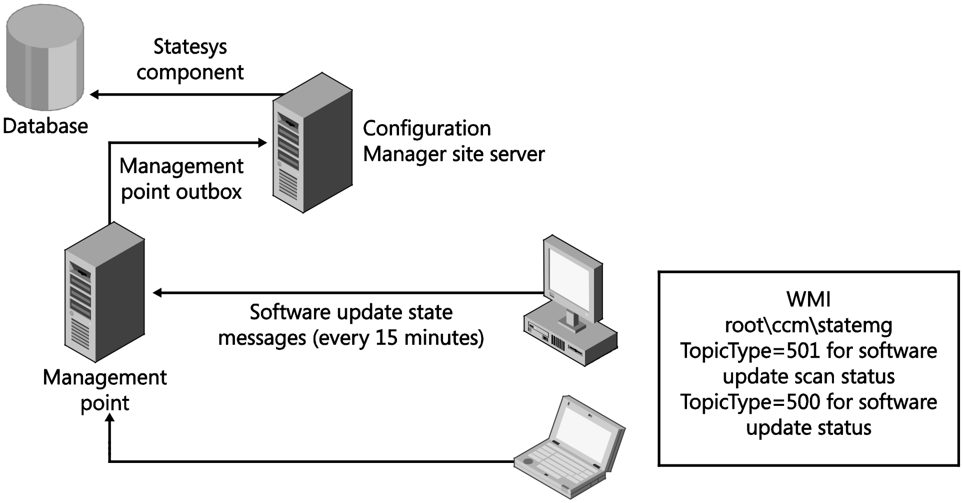
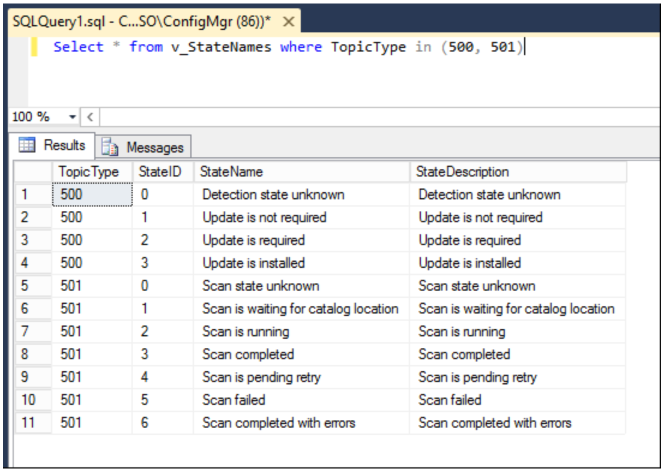

Database
Statesys component
Configuration Manager site server
Management point outbox
Software update state messages (every 15 minutes)
Management point
WMI
root\ccm\statemg
TopicType=501 for software update scan status
TopicType=500 for software update status

FIGURE 4-7 Software update state messages workflow

You can also run the following SQL query in the Configuration Manager database to get the list of state messages and their descriptions, as shown in Figure 4-8:

```
Select * from v_StateNames where TopicType in (500, 501)
```



```
Select * from v_StateNames where TopicType in (500, 501)
```

	TopicType	StateID	StateName	StateDescription
1	500	0	Detection state unknown	Detection state unknown
2	500	1	Update is not required	Update is not required
3	500	2	Update is required	Update is required
4	500	3	Update is installed	Update is installed
5	501	0	Scan state unknown	Scan state unknown
6	501	1	Scan is waiting for catalog location	Scan is waiting for catalog location
7	501	2	Scan is running	Scan is running
8	501	3	Scan completed	Scan completed
9	501	4	Scan is pending retry	Scan is pending retry
10	501	5	Scan failed	Scan failed
11	501	6	Scan completed with errors	Scan completed with errors

FIGURE 4-8 SQL query for state names

Software update summarization

The software update summarization task updates the status of software updates in the console for all clients in the Configuration Manager hierarchy. The summarizer does not provide real-time information in the console because it runs every hour by default, and there are other processes running in the background to collect the information from the SQL database and display it in the console. To initiate the data summarization, in the Software Library workspace, click Software Update, click All Software Updates, and then click Run Summarization on the ribbon.

Software update summarization runs the spTask_SUM_UpdateStatusSummarizer stored procedure in the background. This stored procedure processes updates in a batch of the top 200 configuration items for best performance since less than 100 will impact overall performance while more than 1,000 will make each loop longer and potentially block other processes. It will also exclude from the summarization any configuration items that are already up to date. You can monitor software update status summarization in Statesys.log on the primary site server using the following entries:

```
SQL MESSAGE: spTask_SUM_UpdateStatusSummarizer - 12:36:52:017:
spTask_SUM_UpdateStatusSummarizer started with watermark:0  SMS_STATE_SYSTEM 12/5/2014
12:37:05 PM      6816 (0x1AA0)

SQL MESSAGE: spTask_SUM_UpdateStatusSummarizer - 12:36:52:200: processed to:37114,total
processed: 5     SMS_STATE_SYSTEM 12/5/2014 12:37:05 PM     6816 (0x1AA0)

SQL MESSAGE: spTask_SUM_UpdateStatusSummarizer - 12:36:52:203: processed to:37166,total
processed: 7     SMS_STATE_SYSTEM 12/5/2014 12:37:05 PM     6816 (0x1AA0)

SQL MESSAGE: spTask_SUM_UpdateStatusSummarizer - 12:36:52:327:
spTask_SUM_UpdateStatusSummarizer done. watermark:NULL,total count:7 SMS_STATE_SYSTEM
         12/5/2014 12:37:05 PM     6816 (0x1AA0)
```

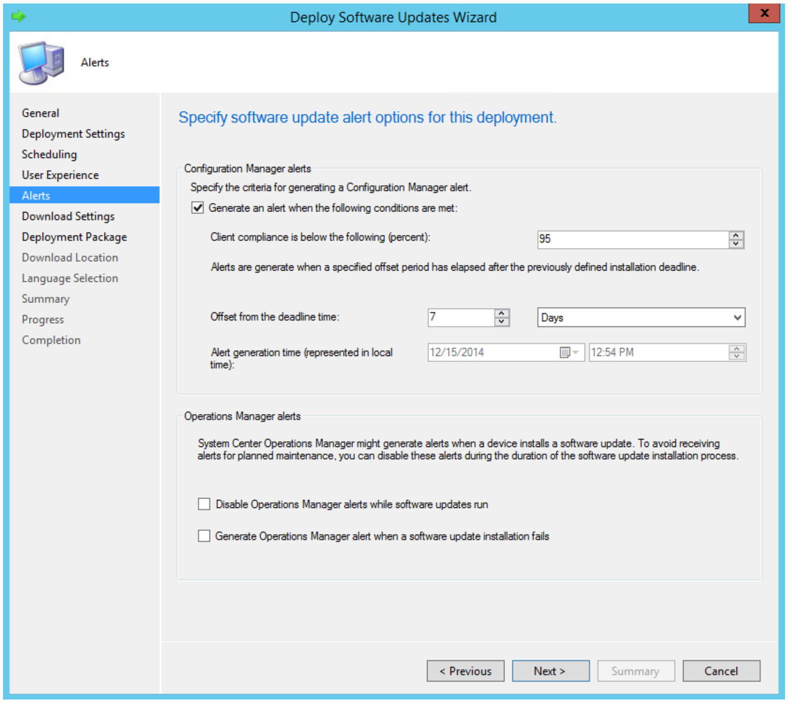
Note how many configuration items have been processed and the time it took to complete in Statesys.log entries. You can confirm the completion of the software update summarizer task by reviewing the entry “spTask_SUM_UpdateStatusSummarizer done” in the same log.

Alerts

The new alerts feature in System Center 2012 R2 Configuration Manager is completely different than Microsoft System Center Operations Manager (SCOM) monitoring. Alerts in Configuration Manager are generated by some operations when a specific condition occurs. You can view and monitor alerts in the Monitoring workspace of the Configuration Manager console. You can also configure subscriptions for selected alerts to be emailed to you automatically. There are different types of alert state in the Configuration Manager console:

- **Active** When the condition of an alert is met, the alert becomes active.
- **Disabled** When an alert is disabled by the Configuration Manager administrator, it does not process any updates and the status of the alert does not change.
- **Never triggered** This alert state means the condition of the alert was never met, so it was never triggered.
- **Postponed** A postponed active alert won’t indicate an evaluation of the state of an alert until a specified date.
- **Canceled** An alert is canceled when its conditions are no longer met, although the alert is still enabled and auto-resolved when the issue is fixed.

You can configure alerts for different objects in Configuration Manager. For software updates, you can set an alert for a low deployment success rate of an update group or individual update deployments. Figure 4-9 shows the Alerts page of the Deploy Software Updates Wizard. You can set an alert if the client compliance percent falls below a certain percentage after a certain deadline. For example, if your deployment is set to run as soon as possible and you set the offset value from the deadline to seven days, the alert generation time will be seven days from the deadline time.



Specify software update alert options for this deployment.

Configuration Manager alerts

Specify the criteria for generating a Configuration Manager alert.

☑ Generate an alert when the following conditions are met:

Client compliance is below the following (percent): 95

Alerts are generate when a specified offset period has elapsed after the previously defined installation deadline.

Offset from the deadline time: 7 Days

Alert generation time (represented in local time): 12/15/2014 12:54 PM

Operations Manager alerts

System Center Operations Manager might generate alerts when a device installs a software update. To avoid receiving alerts for planned maintenance, you can disable these alerts during the duration of the software update installation process.

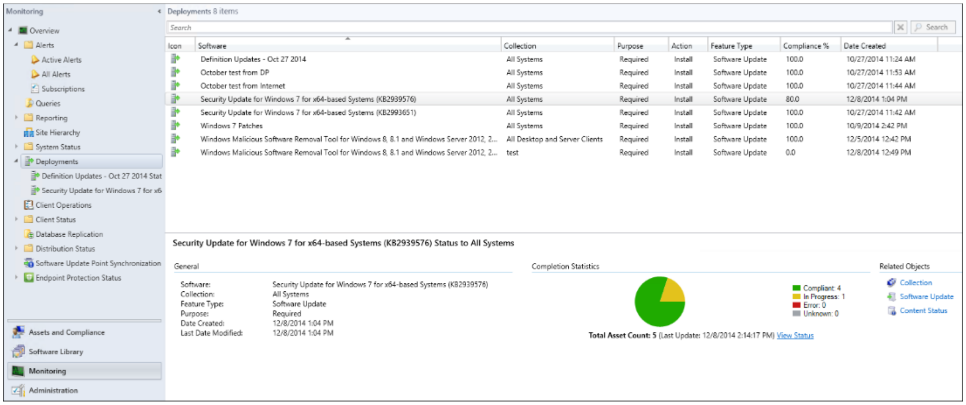
☐ Disable Operations Manager alerts while software updates run

☐ Generate Operations Manager alert when a software update installation fails

FIGURE 4-9 Deploy Software Updates Wizard

Monitoring an individual update

You can monitor an individual update in the Monitoring workspace of the Configuration Manager console. Select the Deployments node in the Monitoring workspace to review all the deployments in the console. This includes all types of deployments, such as application or software updates. Figure 4-10 shows completion statistics for software update KB2939576.



Deployments 8 items

Icon	Software	Collection	Purpose	Action	Feature Type	Compliance %	Date Created
	Definition Updates - Oct 27 2014	All Systems	Required	Install	Software Update	100.0	10/27/2014 11:24 AM
	October test from DP	All Systems	Required	Install	Software Update	100.0	10/27/2014 11:53 AM
	October test from Internet	All Systems	Required	Install	Software Update	100.0	10/27/2014 11:44 AM
	Security Update for Windows 7 for x64-based Systems (KB2939576)	All Systems	Required	Install	Software Update	80.0	12/8/2014 1:04 PM
	Security Update for Windows 7 for x64-based Systems (KB2993651)	All Systems	Required	Install	Software Update	100.0	10/27/2014 11:42 AM
	Windows 7 Patches	All Systems	Required	Install	Software Update	100.0	10/9/2014 2:42 PM
	Windows Malicious Software Removal Tool for Windows 8, 8.1 and Windows Server 2012, 2...	All Desktop and Server Clients	Required	Install	Software Update	100.0	12/5/2014 12:42 PM
	Windows Malicious Software Removal Tool for Windows 8, 8.1 and Windows Server 2012, 2...	test	Required	Install	Software Update	0.0	12/8/2014 12:49 PM

Security Update for Windows 7 for x64-based Systems (KB2939576) Status to All Systems

General

Software: Security Update for Windows 7 for x64-based Systems (KB2939576)
Collection: All Systems
Feature Type: Software Update
Purpose: Required
Date Created: 12/8/2014 1:04 PM
Last Date Modified: 12/8/2014 1:04 PM

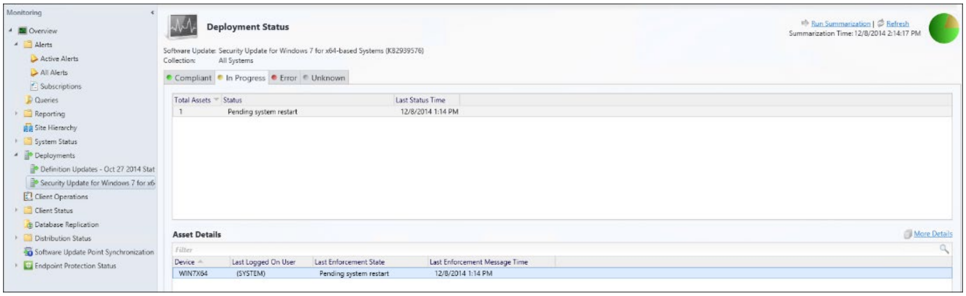
Completion Statistics

🟩 Compliant: 4
🟨 In Progress: 1
🟥 Error: 0
⬜ Unknown: 0

Total Asset Count: 5 (Last Update: 12/8/2014 2:14:17 PM) View Status

FIGURE 4-10 Deployment completion statistics

Click View Status under Completion Statistics to view the compliant status of the deployment. There are four tabs under Deployment Status, as shown in Figure 4-11. Click the In Progress tab to view the systems that are either installing the update or pending system restart. This tab also shows the details of the system, such as device name, last logged on user, last enforcement state, and last enforcement time. Click the More Details link located in lower-right side of the window to open the Asset Message window, which includes two tabs: Details and Software Updates. The Details tab provides a summary of the update, and the Software Updates tab provides additional details of the software updates, such as the article ID, the bulletin ID, and more.



Deployment Status

Summarization Time: 12/8/2014 2:14:17 PM

Software Update: Security Update for Windows 7 for x64-based Systems (KB2939576)
Collection: All Systems

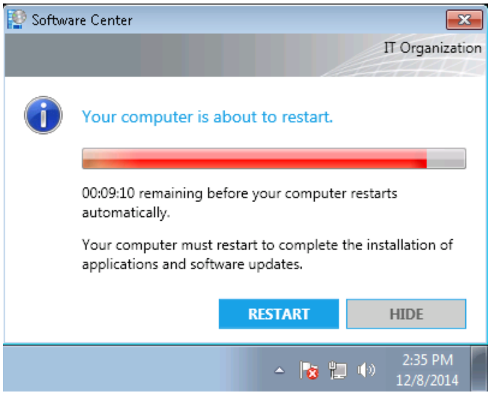
Total Assets	Status	Last Status Time
1	Pending system restart	12/8/2014 1:14 PM

Asset Details

Device	Last Logged On User	Last Enforcement State	Last Enforcement Message Time
WIN7X64	(SYSTEM)	Pending system restart	12/8/2014 1:14 PM

FIGURE 4-11 Deployment status

After the system receives the deployment and installs the update, it then provides notification in the task bar informing the end user that the updates are downloading and installing. If an update requires a reboot, another notification appears with a countdown message, as shown in Figure 4-12. The countdown time depends on the time set in the Configuration Manager site settings.



Your computer is about to restart.

00:09:10 remaining before your computer restarts automatically.

Your computer must restart to complete the installation of applications and software updates.

FIGURE 4-12 Software Center restart countdown notification

Many Configuration Manager client side features are also involved during software update deployment. Figure 4-13 shows the workflow of the main features involved during a software update deployment on the client side.

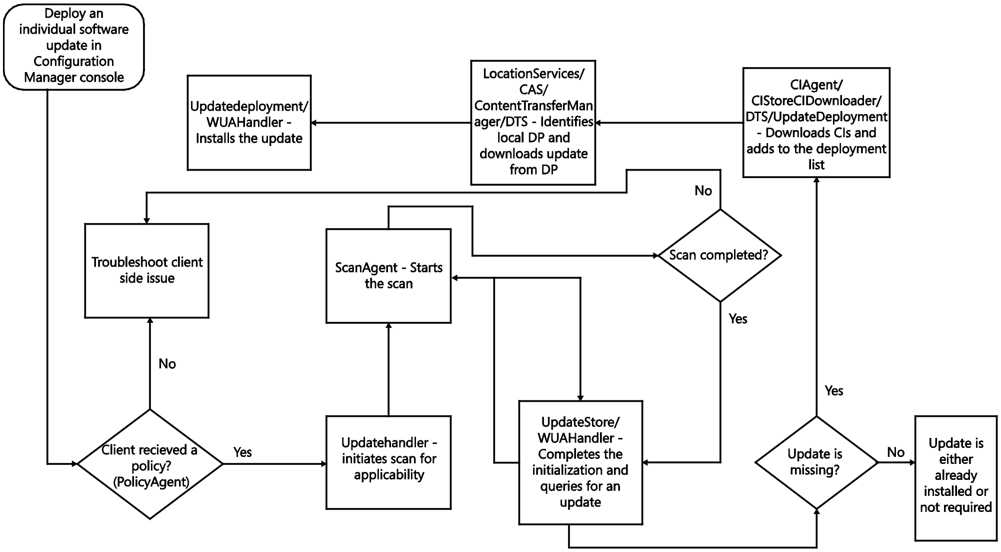
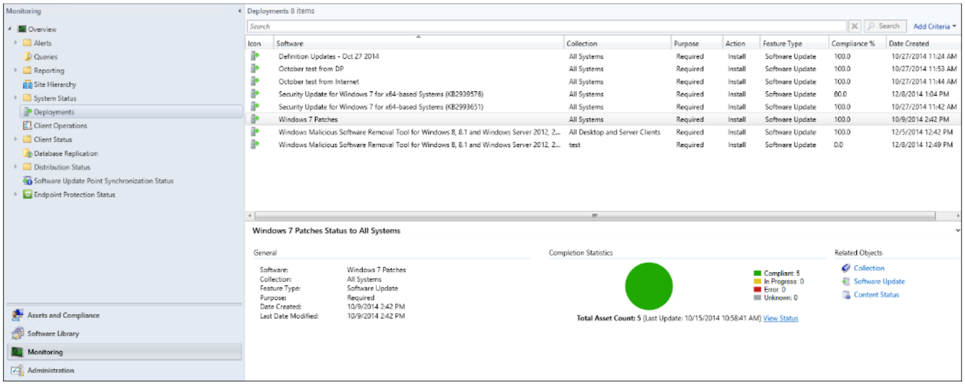

Deploy an individual software update in Configuration Manager console
Updatedeployment/ WUAHandler - Installs the update
LocationServices/ CAS/ ContentTransferManager/DTS - Identifies local DP and downloads update from DP
CIAgent/ CIStoreCIDownloader/ DTS/UpdateDeployment - Downloads CIs and adds to the deployment list
No
Troubleshoot client side issue
ScanAgent - Starts the scan
Scan completed?
Yes
No
Yes
Client recieved a policy? (PolicyAgent)
Yes
Updatehandler - initiates scan for applicability
UpdateStore/ WUAHandler - Completes the initialization and queries for an update
Update is missing?
No
Update is either already installed or not required

FIGURE 4-13 Configuration Manager client side software update deployment flowchart

Monitoring a deployment

Monitoring an entire deployment that includes more than one update is almost the same as monitoring an individual update. You can view the entire deployment by clicking the Deployments node in the Monitoring workspace. Select the deployment on the right side of the console to view completion statistics as described in the previous section, "Monitoring an individual update." Click View Status under Completion Statistics. For example, Figure 4-14 shows the completion statistics for the selected deployment called "Windows 7 Patches." It also shows five systems are compliant out of total asset count five.



Icon	Software	Collection	Purpose	Action	Feature Type	Compliance %	Date Created
	Definition Updates - Oct 27 2014	All Systems	Required	Install	Software Update	100.0	10/27/2014 11:24 AM
	October test from DP	All Systems	Required	Install	Software Update	100.0	10/27/2014 11:53 AM
	October test from Internet	All Systems	Required	Install	Software Update	100.0	10/27/2014 11:44 AM
	Security Update for Windows 7 for x64-based Systems (KB2939576)	All Systems	Required	Install	Software Update	80.0	12/8/2014 1:04 PM
	Security Update for Windows 7 for x64-based Systems (KB2993651)	All Systems	Required	Install	Software Update	100.0	10/27/2014 11:42 AM
	Windows 7 Patches	All Systems	Required	Install	Software Update	100.0	10/9/2014 2:42 PM
	Windows Malicious Software Removal Tool for Windows 8, 8.1 and Windows Server 2012, 2...	All Desktop and Server Clients	Required	Install	Software Update	100.0	12/5/2014 12:42 PM
	Windows Malicious Software Removal Tool for Windows 8, 8.1 and Windows Server 2012, 2...	test	Required	Install	Software Update	0.0	12/8/2014 12:49 PM

Windows 7 Patches Status to All Systems

General

Software: Windows 7 Patches
Collection: All Systems
Feature Type: Software Update
Purpose: Required
Date Created: 10/9/2014 2:42 PM
Last Date Modified: 10/9/2014 2:42 PM

Completion Statistics

🟩 Compliant: 5
🟨 In Progress: 0
🟥 Error: 0
⬜ Unknown: 0

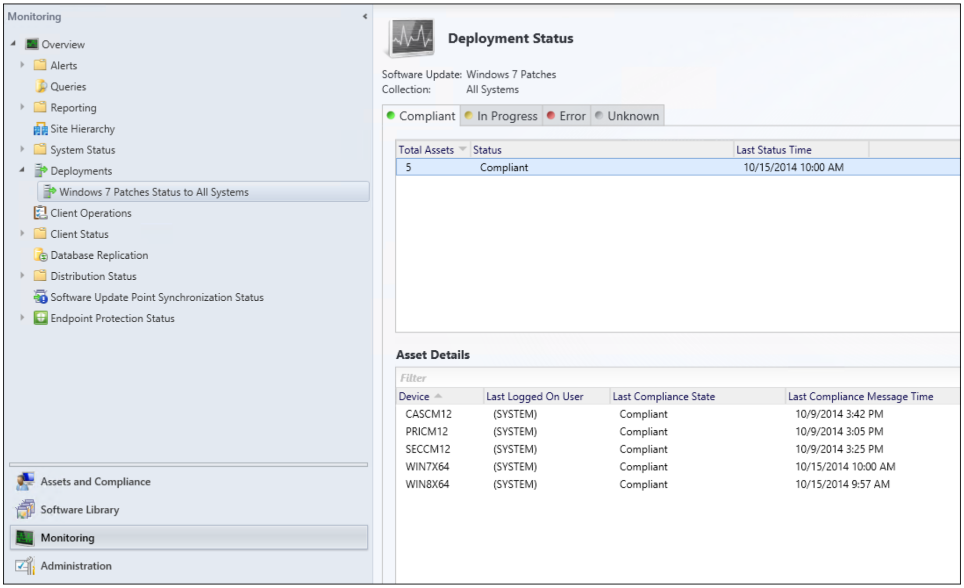
Total Asset Count: 5 (Last Update: 10/15/2014 10:58:41 AM) View Status

Related Objects

Collection
Software Update
Content Status

FIGURE 4-14 Deployment completion statistics for an entire deployment

Click View Status under Completion Statistics to review the overall status of the deployment, including the compliant, in progress, error, and unknown status of the deployment, as well as asset details, including all systems in a specific state (see Figure 4-15).



Software Update: Windows 7 Patches
Collection: All Systems

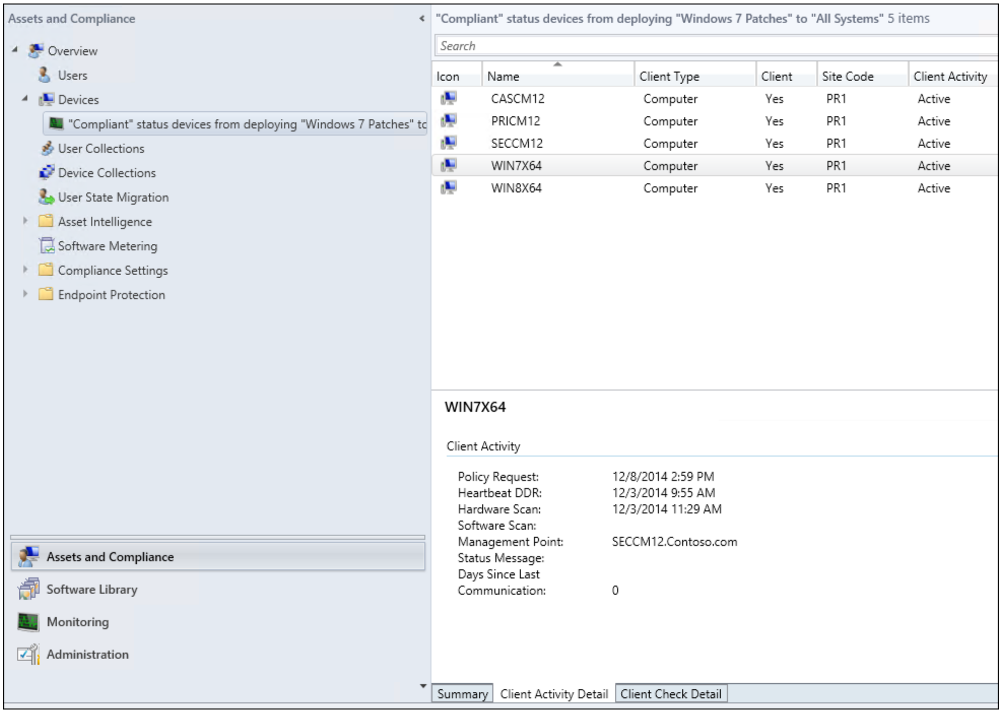
Total Assets	Status	Last Status Time
5	Compliant	10/15/2014 10:00 AM

Asset Details

Device	Last Logged On User	Last Compliance State	Last Compliance Message Time
CASCM12	(SYSTEM)	Compliant	10/9/2014 3:42 PM
PRICM12	(SYSTEM)	Compliant	10/9/2014 3:05 PM
SECCM12	(SYSTEM)	Compliant	10/9/2014 3:25 PM
WIN7X64	(SYSTEM)	Compliant	10/15/2014 10:00 AM
WIN8X64	(SYSTEM)	Compliant	10/15/2014 9:57 AM

FIGURE 4-15 Deployment status of an entire deployment

Double-click the number in the Total Assets column to open another temporary node with a list of systems and their status, as shown in Figure 4-16. Click each system to review its summary, client activity detail, and client check detail and to determine when a client requested a policy, a heartbeat Data Discovery Record (DDR), and a hardware scan.



"Compliant" status devices from deploying "Windows 7 Patches" to "All Systems" 5 items

Icon	Name	Client Type	Client	Site Code	Client Activity
	CASCM12	Computer	Yes	PR1	Active
	PRICM12	Computer	Yes	PR1	Active
	SECCM12	Computer	Yes	PR1	Active
	WIN7X64	Computer	Yes	PR1	Active
	WIN8X64	Computer	Yes	PR1	Active

WIN7X64

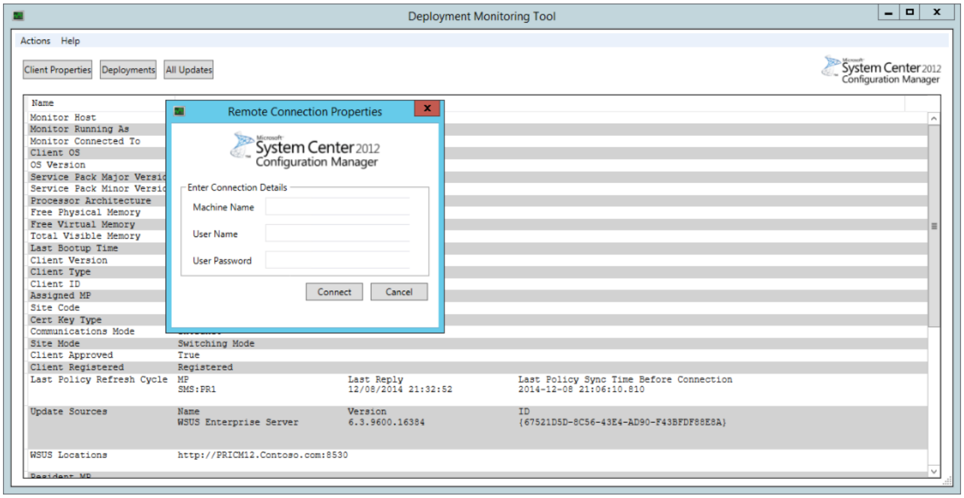
Client Activity

Policy Request:	12/8/2014 2:59 PM
Heartbeat DDR:	12/3/2014 9:55 AM
Hardware Scan:	12/3/2014 11:29 AM
Software Scan:	
Management Point:	SECCM12.Contoso.com
Status Message:	
Days Since Last Communication:	0

FIGURE 4-16 Compliant status devices from deploying Windows 7 Patches to all systems

Deployment Monitoring Tool

The Configuration Manager 2012 R2 toolkit can be used to monitor software updates and other deployments. You can download the toolkit, including the Deployment Monitoring Tool, from *http://www.microsoft.com/en-us/download/details.aspx?id=36213*. This toolkit can be installed on any system running Windows 7 or higher. The Deployment Monitoring Tool is a client side tool that allows an administrator to monitor all deployments on a specific system. When the toolkit is installed, the Deployment Monitoring Tool (DeployMonitoringTool.exe) can be found under the default installation path C:\Program Files (x86)\ConfigMgr 2012 Toolkit R2\ClientTools. By default, the Deployment Monitoring Tool connects to the local machine where the toolkit is installed. To connect to the remote machine and monitor the deployment, click Actions, click Connect To Remote Machine, and then enter the machine name and proper credentials, as shown in Figure 4-17.


Deployment Monitoring Tool
Actions
Help
Client Properties
Deployments
All Updates
Microsoft System Center 2012 Configuration Manager
Remote Connection Properties
Microsoft System Center 2012 Configuration Manager
Enter Connection Details
Machine Name
User Name
User Password
Connect
Cancel

```
Name
Monitor Host
Monitor Running As
Monitor Connected To
Client OS
OS Version
Service Pack Major Versic
Service Pack Minor Versic
Processor Architecture
Free Physical Memory
Free Virtual Memory
Total Visible Memory
Last Bootup Time
Client Version
Client Type
Client ID
Assigned MP
Site Code
Cert Key Type
Communications Mode
Site Mode                  Switching Mode
Client Approved            True
Client Registered          Registered
Last Policy Refresh Cycle  MP                       Last Reply            Last Policy Sync Time Before Connection
                           SMS:PR1                  12/08/2014 21:32:52   2014-12-08 21:06:10.810

Update Sources             Name                     Version               ID
                           WSUS Enterprise Server   6.3.9600.16384        {67521D5D-8C56-43E4-AD90-F43BFDF88E8A}


WSUS Locations             http://PRICM12.Contoso.com:8530

Resident MP
```

FIGURE 4-17 The Deployment Monitoring Tool and the Remote Connection Properties dialog box

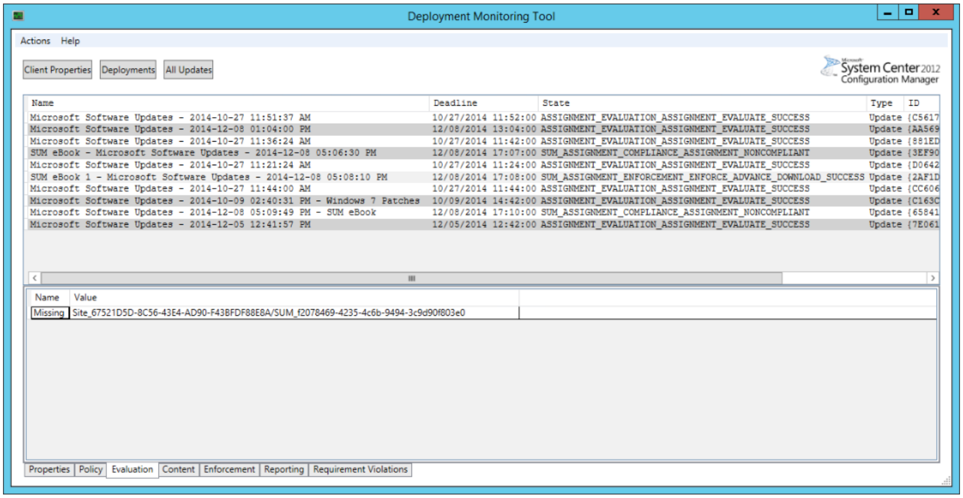
In the Deployment Monitoring Tool, there are three tabs: Client Properties, Deployments, and All Updates.

- **Client Properties** This tab provides information about the machine name, the operating system and its version, physical and virtual memory, last bootup time, last policy refresh cycle, and many other details.
- **Deployments** This tab provides a list of deployments targeted to the system, deadline, state, and type of deployment.
- **All Updates** This tab provides a list of individual updates with article ID, bulletin ID, status and title of the update, and scan time.

There is list of deployments on the Deployments tab, as shown in Figure 4-18. Click an individual deployment to review its properties, policy, evaluation, and other details. In Figure 4-18, the SUM eBook – Microsoft Software Updates - 2014-12-08 05:06:30 PM is highlighted and the Evaluation tab is selected. The current status of the deployment is Missing. If more than one update is associated with the deployment, each update is listed on the Evaluation tab. To determine the bulletin ID, article ID, title, and other details for an update listed in the Value column on the Evaluation tab, use the following SQL query:

```
Select BulletinID, ArticleID, Title, DateRevised, DatePosted, ModelName, CI_ID,
CI_UniqueID from v_UpdateInfo where ModelName = '<insert model name>'
```

> **NOTE** Replace ModelName in the above query with the model name similar to the string 'Site_67521D5D-8C56-43E4-AD90-F43BFDF88E8A/SUM_ced3293c-2613-41ff-bd6a-d8525504c035'.



Name	Deadline	State	Type	ID
Microsoft Software Updates - 2014-10-27 11:51:37 AM	10/27/2014 11:52:00	ASSIGNMENT_EVALUATION_ASSIGNMENT_EVALUATE_SUCCESS	Update	{C5617
Microsoft Software Updates - 2014-12-08 01:04:00 PM	12/08/2014 13:04:00	ASSIGNMENT_EVALUATION_ASSIGNMENT_EVALUATE_SUCCESS	Update	{AA569
Microsoft Software Updates - 2014-10-27 11:36:24 AM	10/27/2014 11:42:00	ASSIGNMENT_EVALUATION_ASSIGNMENT_EVALUATE_SUCCESS	Update	{881ED
SUM eBook - Microsoft Software Updates - 2014-12-08 05:06:30 PM	12/08/2014 17:07:00	SUM_ASSIGNMENT_COMPLIANCE_ASSIGNMENT_NONCOMPLIANT	Update	{3EF90
Microsoft Software Updates - 2014-10-27 11:21:24 AM	10/27/2014 11:24:00	ASSIGNMENT_EVALUATION_ASSIGNMENT_EVALUATE_SUCCESS	Update	{D0642
SUM eBook 1 - Microsoft Software Updates - 2014-12-08 05:08:10 PM	12/08/2014 17:08:00	SUM_ASSIGNMENT_ENFORCEMENT_ENFORCE_ADVANCE_DOWNLOAD_SUCCESS	Update	{2AF1D
Microsoft Software Updates - 2014-10-27 11:44:00 AM	10/27/2014 11:44:00	ASSIGNMENT_EVALUATION_ASSIGNMENT_EVALUATE_SUCCESS	Update	{CC606
Microsoft Software Updates - 2014-10-09 02:40:31 PM - Windows 7 Patches	10/09/2014 14:42:00	ASSIGNMENT_EVALUATION_ASSIGNMENT_EVALUATE_SUCCESS	Update	{C163C
Microsoft Software Updates - 2014-12-08 05:09:49 PM - SUM eBook	12/08/2014 17:10:00	SUM_ASSIGNMENT_COMPLIANCE_ASSIGNMENT_NONCOMPLIANT	Update	{65841
Microsoft Software Updates - 2014-12-05 12:41:57 PM	12/05/2014 12:42:00	ASSIGNMENT_EVALUATION_ASSIGNMENT_EVALUATE_SUCCESS	Update	{7E061

Name	Value
Missing	Site_67521D5D-8C56-43E4-AD90-F43BFDF88E8A/SUM_f2078469-4235-4c6b-9494-3c9d90f803e0

FIGURE 4-18 The Deployment Monitoring Tool showing deployment status

Different states are listed under each of the deployments in the Deployment Monitoring Tool. For example, a state of a deployment might be set to ASSIGNMENT_EVALUATION_ASSIGNMENT_EVALUATE_FAILED. This means that either the client or the server is having an issue evaluating updates. To determine whether it is a server or client side issue, review the client logs located in the CCM\Logs folder, such as WUAHandler.log, ScanAgent.log, UpdateDeployment.log, or WindowsUpdate.log. If many clients are experiencing the same issue, it is most likely a server side problem. If only a few clients are having an issue, it could be related to a client side issue. In any case, logs can provide more details.

On the client side, review the following logs:

```
WUAHandler.log:

OnSearchComplete - Failed to end search job. Error = 0x80072ee2.     WUAHandler
        12/9/2014 4:19:31 PM 1904 (0x0770)

Scan failed with error = 0x80072ee2.       WUAHandler       12/9/2014 4:19:31 PM
        1904 (0x0770)

ScanAgent.log:

ScanJob({1548AD46-F353-46DC-8312-783F2298BD8A}): CScanJob::OnScanComplete -Scan Failed
with Error=0x80072ee2     ScanAgent        12/9/2014 4:20:34 PM      3552 (0x0DE0)

ScanJob({1548AD46-F353-46DC-8312-783F2298BD8A}): CScanJobManager::OnScanComplete- failed
at CScanJob::OnScanComplete with error=0x80072ee2   ScanAgent        12/9/2014 4:20:34 PM
        3552 (0x0DE0)
```

```
WindowsUpdate.log:

2014-12-09       16:20:34:430      124     320      Misc     WARNING: Send failed with hr
= 80072ee2.

2014-12-09       16:20:34:430      124     320      Misc     WARNING: SendRequest failed
with hr = 80072ee2. Proxy List used: <(null)> Bypass List used : <(null)> Auth Schemes
used : <>

2014-12-09       16:20:34:430      124     320      Misc     FATAL: SOAP/WinHttp -
SendRequest: SendRequestUsingProxy failed. error 0x80072ee2

2014-12-09       16:20:34:430      124     320      PT         + Last proxy send request
failed with hr = 0x80072EE2, HTTP status code = 0
```

All of the logs show the same error code, 0x80072ee2. The Error Lookup tool in CMTrace converts the error code into a meaningful name and reveals that it indicates that the operation timed out. This means the client is timing out when trying to scan against the Windows Server Update Services (WSUS) server, so most likely this is a server side issue.

Review the WCM.log on a primary site server.

```
WCM.log on Primary site server:

Verify Upstream Server settings on the Active WSUS Server
        SMS_WSUS_CONFIGURATION_MANAGER 12/9/2014 3:25:29 PM 1152 (0x0480)

System.Data.SqlClient.SqlException (0x80131904): A network-related or instance-specific
error occurred while establishing a connection to SQL Server. The server was not found
or was not accessible. Verify that the instance name is correct and that SQL Server is
configured to allow remote connections. (provider: Named Pipes Provider, error: 40 -
Could not open a connection to SQL Server)~~   at
Microsoft.UpdateServices.Internal.BaseApi.SoapExceptionProcessor.DeserializeAndThrow(Soa
pException soapException)~~   at
Microsoft.UpdateServices.Internal.DatabaseAccess.AdminDataAccessProxy.ExecuteSPGetConfig
uration()~~   at
Microsoft.UpdateServices.Internal.BaseApi.UpdateServerConfiguration.Load()~~   at
Microsoft.UpdateServices.Internal.ClassFactory.CreateWellKnownType(Type type, Object[]
args)~~   at Microsoft.UpdateServices.Internal.ClassFactory.CreateInstance(Type type,
Object[] args)~~   at
Microsoft.UpdateServices.Internal.BaseApi.UpdateServer.GetConfiguration()~~   at
Microsoft.SystemsManagementServer.WSUS.WSUSServer.SetUpstreamServerSettings(Boolean
SyncFromMicrosoftUpdate, Boolean ReplicaServer, String UpstreamWSUSServerName, Int32
UpstreamWSUSServerPortNumber, Boolean UseSSL, Boolean HostBinariesOnMU, Int32
ReportingLevel, Int32 MaximumAllowedComputers)~~ClientConnectionId:00000000-0000-0000-
0000-000000000000SMS_WSUS_CONFIGURATION_MANAGER    12/9/2014 3:26:29 PM     1152
(0x0480)

Remote configuration failed on WSUS Server.        SMS_WSUS_CONFIGURATION_MANAGER
        12/9/2014 3:26:29 PM     1152 (0x0480)
```

Based on the WCM.log, the issue seems to be related to SQL Server. Verify that the SQL Server service (MSSQLSERVER) is up and running and that you can connect to the SQL database. Next make sure WSUS is up and running. In this case, it turned out that the WSUS service on the primary site server was stopped, so the client could not scan against its WSUS server. After the WSUS service was started, the software update scan started working on the client.

Built-in and custom reports

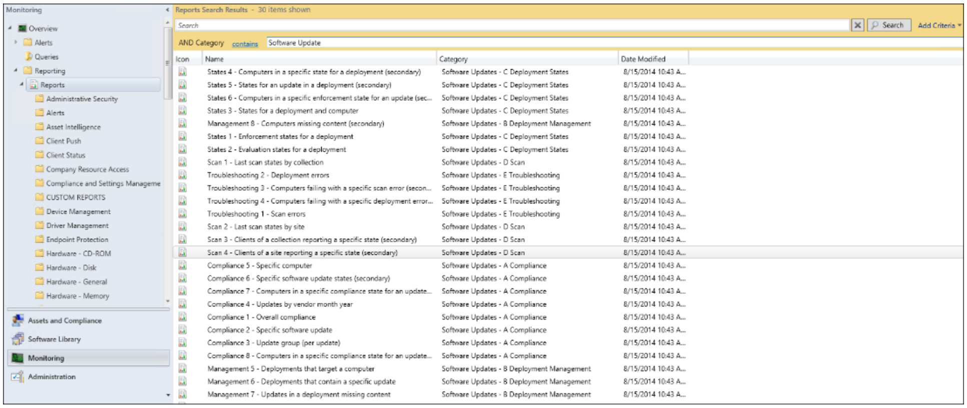
Microsoft System Center 2012 R2 Configuration Manager includes a number of different built-in reports that you can use to retrieve information from the Configuration Manager database. These reports provide great visibility into your overall environment. Configuration Manager uses Microsoft SQL Server as its backend database engine. During hardware and software scans, Configuration Manager scans the local machine and sends the information collected to the backend database to be stored there. Since the data collected resides in a SQL database, Transact-SQL (T-SQL) queries can be used to query the database and retrieve information about the data stored in it.

There is a total of 469 built-in reports in the Configuration Manager console. Of these, 30 built-in reports are just for software updates. These reports are helpful for troubleshooting any kind of issue related to software updates. There are specific reports just for software update troubleshooting as well.

Software update reports

To run and view software update reports, complete the following steps:

1. In the Configuration Manager console, in the Monitoring workspace, click Reporting.
2. Click Reports. All of the available reports display on the right side of the console.
3. Click the Add Criteria field, select Category, and then select Add.
4. In the Category Contains field, type **Software Update**, and then click Search to display all reports related to software updates, as shown in Figure 4-19.



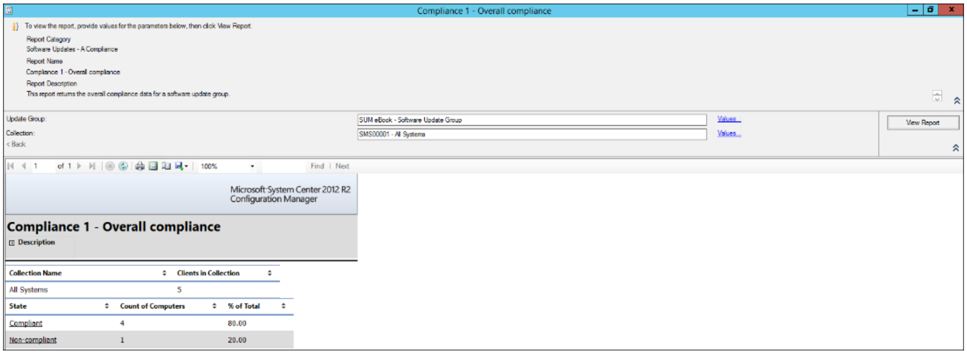
Reports Search Results - 30 items shown

AND Category contains Software Update

Icon	Name	Category	Date Modified
	States 4 - Computers in a specific state for a deployment (secondary)	Software Updates - C Deployment States	8/15/2014 10:43 A...
	States 5 - States for an update in a deployment (secondary)	Software Updates - C Deployment States	8/15/2014 10:43 A...
	States 6 - Computers in a specific enforcement state for an update (sec...	Software Updates - C Deployment States	8/15/2014 10:43 A...
	States 3 - States for a deployment and computer	Software Updates - C Deployment States	8/15/2014 10:43 A...
	Management 8 - Computers missing content (secondary)	Software Updates - B Deployment Management	8/15/2014 10:43 A...
	States 1 - Enforcement states for a deployment	Software Updates - C Deployment States	8/15/2014 10:43 A...
	States 2 - Evaluation states for a deployment	Software Updates - C Deployment States	8/15/2014 10:43 A...
	Scan 1 - Last scan states by collection	Software Updates - D Scan	8/15/2014 10:43 A...
	Troubleshooting 2 - Deployment errors	Software Updates - E Troubleshooting	8/15/2014 10:43 A...
	Troubleshooting 3 - Computers failing with a specific scan error (secon...	Software Updates - E Troubleshooting	8/15/2014 10:43 A...
	Troubleshooting 4 - Computers failing with a specific deployment error...	Software Updates - E Troubleshooting	8/15/2014 10:43 A...
	Troubleshooting 1 - Scan errors	Software Updates - E Troubleshooting	8/15/2014 10:43 A...
	Scan 2 - Last scan states by site	Software Updates - D Scan	8/15/2014 10:43 A...
	Scan 3 - Clients of a collection reporting a specific state (secondary)	Software Updates - D Scan	8/15/2014 10:43 A...
	Scan 4 - Clients of a site reporting a specific state (secondary)	Software Updates - D Scan	8/15/2014 10:43 A...
	Compliance 5 - Specific computer	Software Updates - A Compliance	8/15/2014 10:43 A...
	Compliance 6 - Specific software update states (secondary)	Software Updates - A Compliance	8/15/2014 10:43 A...
	Compliance 7 - Computers in a specific compliance state for an update...	Software Updates - A Compliance	8/15/2014 10:43 A...
	Compliance 4 - Updates by vendor month year	Software Updates - A Compliance	8/15/2014 10:43 A...
	Compliance 1 - Overall compliance	Software Updates - A Compliance	8/15/2014 10:43 A...
	Compliance 2 - Specific software update	Software Updates - A Compliance	8/15/2014 10:43 A...
	Compliance 3 - Update group (per update)	Software Updates - A Compliance	8/15/2014 10:43 A...
	Compliance 8 - Computers in a specific compliance state for an update...	Software Updates - A Compliance	8/15/2014 10:43 A...
	Management 5 - Deployments that target a computer	Software Updates - B Deployment Management	8/15/2014 10:43 A...
	Management 6 - Deployments that contain a specific update	Software Updates - B Deployment Management	8/15/2014 10:43 A...
	Management 7 - Updates in a deployment missing content	Software Updates - B Deployment Management	8/15/2014 10:43 A...

FIGURE 4-19 Software update reports

5. Sort by the Name column, and select the first report, Compliance 1 – overall compliance.
6. Right-click Compliance 1 – Overall Compliance and select Run.
7. Specify the Update Group and Collection values, and then click View Report to display overall compliance, as shown in Figure 4-20 for the SUM eBook – Software Update Group.



To view the report, provide values for the parameters below, then click View Report.

Report Category
Software Updates - A Compliance
Report Name
Compliance 1 - Overall compliance
Report Description
This report returns the overall compliance data for a software update group.

Update Group: SUM eBook - Software Update Group
Collection: SMS00001 - All Systems

Compliance 1 - Overall compliance

Collection Name	Clients in Collection
All Systems	5

State	Count of Computers	% of Total
Compliant	4	80.00
Non-compliant	1	20.00

FIGURE 4-20 Overall compliance report

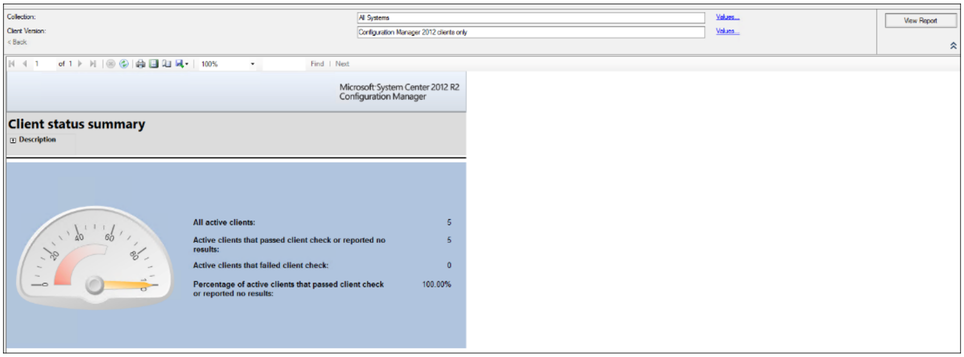
The Overall Compliance report shows the percentage of compliant and non-compliant systems, count of computers in each category, and total clients in the collection you selected. Click either Compliant or Non-compliant state to drill down in the report and open another report called Compliance 7 – Computers in a specific compliance state for an update group (secondary). This report lists the computer names, assigned site, and client version for the state you selected. Select the computer name to open the report called Compliance 5 – Specific Computer, which displays all software updates that are either required or installed on the computer you selected, so you can determine which updates are not installed and take appropriate action.

You can also review the Scan 1 – Last Scan States By Collection report located under the Software Update – D scan node to determine how many clients are running into a software update scan issue. Another useful report for tracking software update deployment is Management 3 – Updates In A Deployment. This report asks for the deployment name and returns all of the updates that are part of the deployment along with the number of updates installed, required, not required, unknown, failed, and pending.

Client status reports

To view and run client status reports, complete the following steps:

1. In the Configuration Manager console, in the Monitoring workspace, click Reporting.
2. Expand Reports, and then click Client Status.
3. On the right side of the console, right-click Client Status Summary Report and select Run.
4. Specify the collection value and the client version, and then click View Report to display a graphic view of active clients that passed client check or reported no results, as shown in Figure 4-21



Collection: All Systems

Client Version: Configuration Manager 2012 clients only

Microsoft System Center 2012 R2
Configuration Manager

Client status summary

All active clients:	5
Active clients that passed client check or reported no results:	5
Active clients that failed client check:	0
Percentage of active clients that passed client check or reported no results:	100.00%

FIGURE 4-21 Client status summary report

You can also review other useful reports such as Clients With Failed Client Check Details, Inactive Client Details, and Client Remediation Summary.

Custom reports

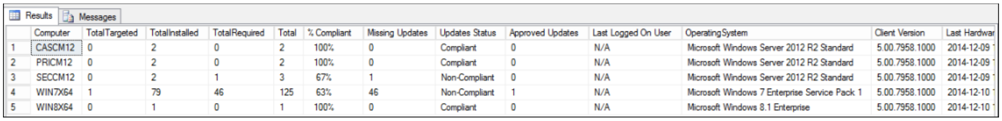
Configuration Manager creates several database tables and views during the site server installation. These views and tables are queried by built-in queries to generate the reports. If the default reports are not sufficient for your needs, however, you can generate custom reports by creating custom T-SQL queries. If you are familiar with SQL queries then it is not difficult to create a custom report using Report Builder. This section discusses the underlying tables and views used for querying the database and generating custom reports.

This section features a couple of sample customer reports and how to create them. For step-by-step details on how to create custom reports in Configuration Manager, download the free ebook *Microsoft System Center: Configuration Manager Field Experience* from *http://blogs.msdn.com/b/microsoft_press/archive/2013/10/03/free-ebook-2-in-this-series-microsoft-system-center-configuration-manager-field-experience.aspx* and review Chapter 2 and Chapter 4.

When you want to determine how many computers in your environment are vulnerable and non-compliant, a single built-in report may not provide all the details you need. Since you may not be approving all the updates Microsoft releases every month, you'll want to make sure computers are compliant against the software update you approved within the Configuration Manager console. The following SQL query provides a list of computers, total targeted updates, total installed, total required, percent compliant, number of missing updates, and update status, as shown in Figure 4-22.

```
SELECT
        rs.Netbios_name0 AS 'Computer',
SUM(CASE WHEN UCS.status=2 and ctm.ResourceID IS NOT NULL THEN 1 ELSE 0 END) AS
TotalTargeted,
SUM(CASE WHEN UCS.status=3 THEN 1 ELSE 0 END) AS TotalInstalled,
SUM(CASE WHEN UCS.status=2 THEN 1 ELSE 0 END) AS TotalRequired,
SUM(CASE WHEN ((UCS.status=2) or (UCS.status=3)) THEN 1 ELSE 0 END) AS Total,
   (STR((SUM(CASE WHEN UCS.status=3 THEN 1 ELSE 0 END) *100.0/SUM(CASE WHEN
((UCS.status=2) or              (UCS.status=3)) THEN 1 ELSE 0 END) ),5)) + '%' AS '%
Compliant',
SUM(CASE WHEN UCS.status=2 THEN 1 ELSE 0 END) AS 'Missing Updates',
    CASE
    WHEN (SUM(CASE WHEN UCS.status=2 THEN 1 ELSE 0 END)) > 0 THEN 'Non-Compliant'
    ELSE 'Compliant'
    END AS 'Updates Status',
SUM(CASE WHEN UCS.status=2 and ctm.ResourceID IS NOT NULL THEN 1 ELSE 0 END) AS
'Approved Updates',
    CASE
         WHEN gcs.UserName0 IS NULL THEN 'N/A'
         ELSE gcs.UserName0
        END AS 'Last Logged On User',
        CASE
           WHEN os.CSDVersion0 IS NULL THEN os.caption0
           WHEN os.caption0 IS NULL THEN 'N/A'
           ELSE os.caption0 + ' ' + os.CSDVersion0
        END AS OperatingSystem,
    rs.Client_Version0 AS 'Client Version',
    ws.lasthwscan AS 'Last Hardware Scan',
    uss.LastScanPackageLocation AS 'Last Scan Location',
    uss.LastScanPackageVersion AS 'Last Scan Pakage',
         st.StateName AS 'Status',
          DATEDIFF(D, OS.LastBootUpTime0, GETDATE()) 'Last Boot (Days)'

FROM
      v_ClientCollectionMembers ccm
      LEFT JOIN v_R_System rs on rs.ResourceID = ccm.ResourceID
      LEFT JOIN v_UpdateComplianceStatus  UCS on UCS.ResourceID = ccm.ResourceID
      LEFT OUTER JOIN v_CITargetedMachines ctm on ctm.CI_ID=UCS.CI_ID and ctm.ResourceID
= rs.ResourceID
      INNER JOIN v_GS_COMPUTER_SYSTEM GCS on GCS.ResourceID = rs.ResourceID
      JOIN v_UpdateInfo ui on ui.CI_ID=UCS.CI_ID
      JOIN v_CICategories_All catall2 on catall2.CI_ID=ui.CI_ID
      JOIN v_CategoryInfo catinfo2 on catall2.CategoryInstance_UniqueID =
catinfo2.CategoryInstance_UniqueID and catinfo2.CategoryTypeName='UpdateClassification'
      INNER JOIN v_gs_workstation_status ws on ws.resourceid=rs.resourceid
      INNER JOIN v_UpdateScanStatus uss on ws.resourceid = uss.ResourceID
      LEFT JOIN v_StateNames st on st.TopicType = 501 and st.StateID = (CASE WHEN
(ISNULL(uss.LastScanState, 0)=0 and Left(ISNULL(rs.Client_Version0, '4.0'),
      1) < '4') THEN 7 ELSE ISNULL(uss.LastScanState, 0) END)
      JOIN v_Gs_Operating_System OS on ws.resourceid = OS.ResourceID

WHERE
      ccm.CollectionID = '<REPLACE WITH Collection ID>'


GROUP BY
      rs.Netbios_name0,
      gcs.UserName0,
      rs.Client_Version0,
      ws.lasthwscan,
      uss.LastScanPackageLocation,
      uss.LastScanPackageVersion ,
      st.StateName,
      OS.LastBootUpTime0,
      os.caption0,
      os.CSDVersion0

      ORDER BY [Computer]
```

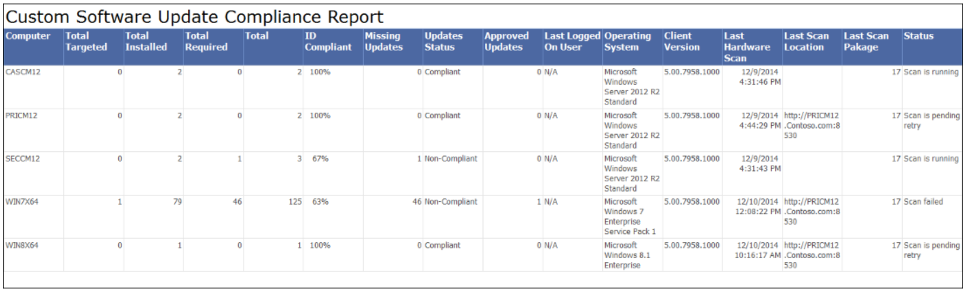
NOTE Replace CollectionID with the collection ID from your Configuration Manager environment.



	Computer	TotalTargeted	TotalInstalled	TotalRequired	Total	% Compliant	Missing Updates	Updates Status	Approved Updates	Last Logged On User	OperatingSystem	Client Version	Last Hardwar
1	CASCM12	0	2	0	2	100%	0	Compliant	0	N/A	Microsoft Windows Server 2012 R2 Standard	5.00.7958.1000	2014-12-09
2	PRICM12	0	2	0	2	100%	0	Compliant	0	N/A	Microsoft Windows Server 2012 R2 Standard	5.00.7958.1000	2014-12-09
3	SECCM12	0	2	1	3	67%	1	Non-Compliant	0	N/A	Microsoft Windows Server 2012 R2 Standard	5.00.7958.1000	2014-12-09
4	WIN7X64	1	79	46	125	63%	46	Non-Compliant	1	N/A	Microsoft Windows 7 Enterprise Service Pack 1	5.00.7958.1000	2014-12-10
5	WIN8X64	0	1	0	1	100%	0	Compliant	0	N/A	Microsoft Windows 8.1 Enterprise	5.00.7958.1000	2014-12-10

FIGURE 4-22 SQL query results of a custom software update report

You can use the same SQL query with Report Builder to create custom reports for Configuration Manager. You can save an .RDL file generated from Report Builder and import it into Configuration Manager SQL Server Reporting Services (SSRS) Point *http://<ConfigMgr Server name>/Reports* so you can access it directly from the web browser, as shown in Figure 4-23.



Custom Software Update Compliance Report

Computer	Total Targeted	Total Installed	Total Required	Total	ID Compliant	Missing Updates	Updates Status	Approved Updates	Last Logged On User	Operating System	Client Version	Last Hardware Scan	Last Scan Location	Last Scan Pakage	Status
CASCM12	0	2	0	2	100%	0	Compliant	0	N/A	Microsoft Windows Server 2012 R2 Standard	5.00.7958.1000	12/9/2014 4:31:46 PM		17	Scan is running
PRICM12	0	2	0	2	100%	0	Compliant	0	N/A	Microsoft Windows Server 2012 R2 Standard	5.00.7958.1000	12/9/2014 4:44:29 PM	http://PRICM12.Contoso.com:8530	17	Scan is pending retry
SECCM12	0	2	1	3	67%	1	Non-Compliant	0	N/A	Microsoft Windows Server 2012 R2 Standard	5.00.7958.1000	12/9/2014 4:31:43 PM		17	Scan is running
WIN7X64	1	79	46	125	63%	46	Non-Compliant	1	N/A	Microsoft Windows 7 Enterprise Service Pack 1	5.00.7958.1000	12/10/2014 12:08:22 PM	http://PRICM12.Contoso.com:8530	17	Scan failed
WIN8X64	0	1	0	1	100%	0	Compliant	0	N/A	Microsoft Windows 8.1 Enterprise	5.00.7958.1000	12/10/2014 10:16:17 AM	http://PRICM12.Contoso.com:8530	17	Scan is pending retry

FIGURE 4-23 Custom software update compliance report results

In the field, different customers have different business requirements for monitoring software update compliance in the Configuration Manager environment. For example, one customer may want a custom report that provides the following information:

- Number of updates required for the computers in a specific collection
- Number of critical and non-critical updates
- Computers pending reboot

The following custom SQL query can be used to create a report using SSRS. The query can be customized so that administrators can use a drop-down menu in the report to select the appropriate collection ~~to~~ compare to hardcoding in the query.

```
--Remove previous temporary table if exists
IF OBJECT_ID(N'TempDB.DBO.#temp_RequiredCollectionMembers') IS NOT NULL
BEGIN
        DROP TABLE #temp_RequiredCollectionMembers
END

--Create table to store key machine fields
CREATE TABLE #temp_RequiredCollectionMembers(
        [ResourceID] [int] NOT NULL,
        [Name] [nvarchar](50) NULL,
        [Last Uptime] [datetime] NULL,
        [Pending Reboot] [nvarchar](3) NOT NULL,
        PRIMARY KEY (ResourceID)
        --CONSTRAINT RequiredCollectionMembers_ResourceID_PK PRIMARY KEY NONCLUSTERED
(ResourceID)
)

--Insert query results into Temporary table
INSERT INTO #temp_RequiredCollectionMembers
SELECT DISTINCT
        ColMembers.ResourceID,
        ColMembers.Name,
        OS.LastBootUpTime0 'Last Uptime',
        CASE
                WHEN UIReboot.LastEnforcementMessageID IS NULL THEN 'No'
                ELSE 'Yes'
        END AS 'Pending Reboot'

FROM v_FullCollectionMembership as ColMembers
        LEFT JOIN v_GS_OPERATING_SYSTEM AS OS ON
                ColMembers.ResourceID = OS.ResourceID
        LEFT JOIN (SELECT DISTINCT UCSR.ResourceID, UCSR.LastEnforcementMessageID
                                FROM v_Update_ComplianceStatusReported AS UCSR
                                WHERE UCSR.LastEnforcementMessageID in (5,9)) AS
UIReboot ON
                                        ColMembers.ResourceID = UIReboot.ResourceID

WHERE ColMembers.ResourceType = 5 and
                ColMembers.CollectionID = '<Collection ID>'

--Remove previous temporary table if exists
IF OBJECT_ID(N'TempDB.DBO.#temp_UpdatesRequired') IS NOT NULL
BEGIN
        DROP TABLE #temp_UpdatesRequired
END

--Create table to store key update information per machine
CREATE TABLE #temp_UpdatesRequired(
        [ResourceID] [int] NOT NULL,
        [New Status] [nvarchar](50) NULL,
        [CI_ID] [int] NOT NULL,
        [UpdateClass] [nvarchar] (255) NULL,
        [SeverityLevel] [nvarchar](255) NOT NULL
)

--Insert query results into Temporary table
 INSERT INTO #temp_UpdatesRequired
 SELECT
```

```
        UCSA.ResourceID,
        CASE UCSA.STATUS
                WHEN '0' THEN 'Unknown'
                WHEN '2' THEN 'Required'
        END AS 'New Status',
        UI.CI_ID,
        UC.CategoryInstanceName as UpdateClass,
        Case UI.Severity
                WHEN '10' THEN 'Critical'
                WHEN '8' THEN 'Important'
                WHEN '6' THEN 'Moderate'
                WHEN '2' THEN 'Low'
                WHEN '0' THEN 'None'
                ELSE 'Unknown'
        END AS SeverityLevel

 FROM v_Update_ComplianceStatusReported UCSA
        INNER JOIN v_UpdateInfo UI ON
                UCSA.CI_ID = UI.CI_ID
        INNER JOIN (SELECT CI_ID, CategoryInstanceName
                                        FROM v_CICategoryInfo_All
                                        WHERE CategoryTypeName = 
'UpdateClassification') as UC ON
                                                UI.CI_ID = UC.CI_ID

   WHERE  ucsa.Status = 2 AND
                ui.CIType_ID in (1,8) AND
                ui.IsHidden=0 and
                UI.IsExpired=0 and
                ui.IsSuperseded=0

CREATE NONCLUSTERED INDEX [#temp_UpdatesRequired_IDX1] ON [#temp_UpdatesRequired]
(
        [ResourceID] ASC
)
CREATE NONCLUSTERED INDEX [#temp_UpdatesRequired_IDX2] ON [#temp_UpdatesRequired]
(
        [UpdateClass] ASC
)
CREATE NONCLUSTERED INDEX [#temp_UpdatesRequired_IDX3] ON [#temp_UpdatesRequired]
(
        [SeverityLevel] ASC
)
CREATE NONCLUSTERED INDEX [#temp_UpdatesRequired_IDX4] ON [#temp_UpdatesRequired]
(
        [ResourceID] ASC,
        [UpdateClass] ASC,
        [SeverityLevel] ASC
)



--Produce results query
SELECT
        Results.Name,
        -- Results.[Last Uptime],
        Results.[Pending Reboot],
        --updates.ArticleID,
        --updates.Title,
        COUNT(*) AS 'Total Patches Required',
        SUM(case when UpdateClass = 'Critical Updates' then 1 when SeverityLevel =
'Critical' then 1 else 0 end) as 'Critical Patches Required',
        SUM(case when UpdateClass = 'Critical Updates' then 0 when SeverityLevel =
'Critical' then 0 else 1 end ) as 'Non-Critical Patches Required'



FROM #temp_RequiredCollectionMembers as Results
        INNER JOIN #temp_UpdatesRequired as Updates ON
                Results.ResourceID = Updates.ResourceID

GROUP BY
        Results.Name,
        Results.[Last Uptime],
        Results.[Pending Reboot]
```

NOTE Replace <Collection ID> with appropriate CollectionID.

This page intentionally left blank

CHAPTER 5

Software updates automation

This chapter covers some of the methods to automate various tasks within the Software Updates feature of System Center 2012 Configuration Manager. Through the use of automation to deploy software updates, significant time and money savings can be realized. There are a number of predictably recurring software update deployment activities and maintenance tasks that can be automated. Automation within any technology typically translates into administrative overhead savings, and therefore into expenditure savings within the IT organization. Automatic deployment rules can be used to systematically create and maintain software update groups, dynamically create deployments meeting specific criteria using defined options on a recurring schedule, and maintain the software update database. Windows PowerShell is a powerful tool that can and should be utilized for automating additional tasks, such as keeping the Windows Server Update Services (WSUS) database optimized, as well as creating, modifying, deleting, or updating software update groups, deployments, and packages. Implementing automation within the Software Updates feature of Configuration Manager can ultimately reduce administrative workload, increase deployment efficiency, and reduce the potential for human error.

Understanding automatic deployment rules

Automatic Deployment Rules is a new feature that was first introduced in the RTM version of System Center 2012 Configuration Manager. By providing administrators with the ability to design a customizable and automated solution to deliver software updates, automatic deployment rules can significantly reduce the amount of administrative overhead required during the monthly software update deployment process. Automatic deployment rules automatically download software updates on a recurring interval from as often as once per hour to as little as once per year, using a fully customizable set of search criteria. Automatic deployment rules are most commonly used to deploy monthly software updates (commonly referred to as "Patch Tuesday") as well as System Center Endpoint Protection definition updates.

When an automatic deployment rule runs, the updates that meet an administratively defined set of criteria are added to either an existing software update group or to a new software update group, which is created automatically by the rule. In the Create Automatic Deployment Rule Wizard, if an existing software update group is selected, all existing updates that currently exist in the group from the previous execution of each automatic deployment rule are removed prior to adding the newly matched updates. The automatic deployment rule then automatically downloads all associated software update content for each of the updates into the specified software update package, distributes the package to the assigned distribution points, and creates a required deployment to the target collection specified in the wizard. An automatic deployment rule can also be configured to create the deployment in a disabled state so that the deployment can be reviewed for accuracy and manually enabled once it is confirmed to meet the desired outcome. While the deployment is disabled, the disabled deployment can also be used to report on compliance in the target collection.

The criteria that is configurable within an automatic deployment rule is as follows:

- The target collection to deploy the updates to
- Whether to automatically enable the deployment or leave it disabled
- The criteria of the software updates to include in the deployment
- The evaluation and deployment schedules for the deployment
- The user experience for the deployment
- The download settings for the content included in the deployment

NOTE **You cannot create a software update group manually and then create an automatic deployment rule to add new updates to the manually created group.**

Creating automatic deployment rules

To create a new automatic deployment rule, right-click the Automatic Deployment Rules node in the Software Library located under the Software Updates node within the Configuration Manager console. The Create Automatic Deployment Rule Wizard takes you through each of the settings and optionally allows you to save the settings as a deployment template to be used when creating additional automatic deployment rules in the future.

The first step in the Create Automatic Deployment Rule Wizard is to provide a name and, optionally, a description for the automatic deployment rule. Next, a previously created and saved template or a built-in template can be used to automatically define a set of criteria for the automatic deployment rule. System Center Configuration Manager 2012 R2 comes with two built-in templates: Definition Updates and Patch Tuesday. The Microsoft endpoint definition pattern release cycle takes place at a frequency of three times per day. Therefore, setting the software update point synchronization to run every eight hours will ensure definitions are deployed to clients in the timeliest frequency possible.

The Patch Tuesday automatic deployment rule template provides a pre-defined template to be used for automatically deploying monthly security updates, which are regularly released on the second Tuesday of each month. This template creates a new software update group each time the script runs and includes all security updates released within the previous day.

Therefore, it is intended to be run within 24 hours of the Patch Tuesday updates being released. This template creates a new software update group each time it runs due to the supported limit of 1,000 software updates for a single software update deployment. Each time an automatic deployment rule runs, the updates that previously existed in the rule at the time of the previous successful run of the rule are removed so that the deployment will never exceed this supported limit.

The Definition Updates template is designed to be utilized for the automatic deployment of System Center Endpoint Protection definition updates. This template adds to an existing software update group. The primary reason for this difference is that definition updates are released and downloaded as often as three times per day, which would cause a significant number of software update groups to be created. Also, the previously released endpoint definitions are more frequently superseded and expired, so these updates will automatically be purged depending on the supersedence behavior configuration defined in the Site Configuration, Sites, Configure Site Components, Software Update Point on the Supersedence Rules tab. Review Table 5-1 for the pre-defined settings within each template.

TABLE 5-1 Built-in automatic deployment rule templates

TEMPLATE SETTING	DEFINITION UPDATES	PATCH TUESDAY
Create new or Use existing software update group	Add to an existing software update group	Create a new software update group
Update state message detail	Only error messages	Success and error messages
Search criteria and classification	Forefront Endpoint Protection 2010, Definition Updates	Security updates released within the last day
Rule evaluation schedule	Run after each software update sync	Run every 30 days
Deployment evaluation time zone	UTC	Client Local Time
Deployment scheduling	Available after one hour; required as soon as possible	Available after four hours; required seven days later
User visual experience	Hide all user notifications	Display in Software Center and show all notifications
Deadline behavior	Install regardless of maintenance window	Install only inside maintenance window
Deployment alerting	Do not generate alert	Generate alert when compliance is below 90% after installation deadline plus seven days
Download and execute behavior	Download on fast and slow boundaries	Download on fast and slow boundaries
Fallback behavior	Fall back to Microsoft Update if not available on Distribution Point	Fall back to Microsoft Update if not available on Distribution Point

See also *For more information, see the description of Forefront endpoint security definition updates at http://technet.microsoft.com/en-us/library/jj822983.aspx and http://support.microsoft.com/kb/977939.*

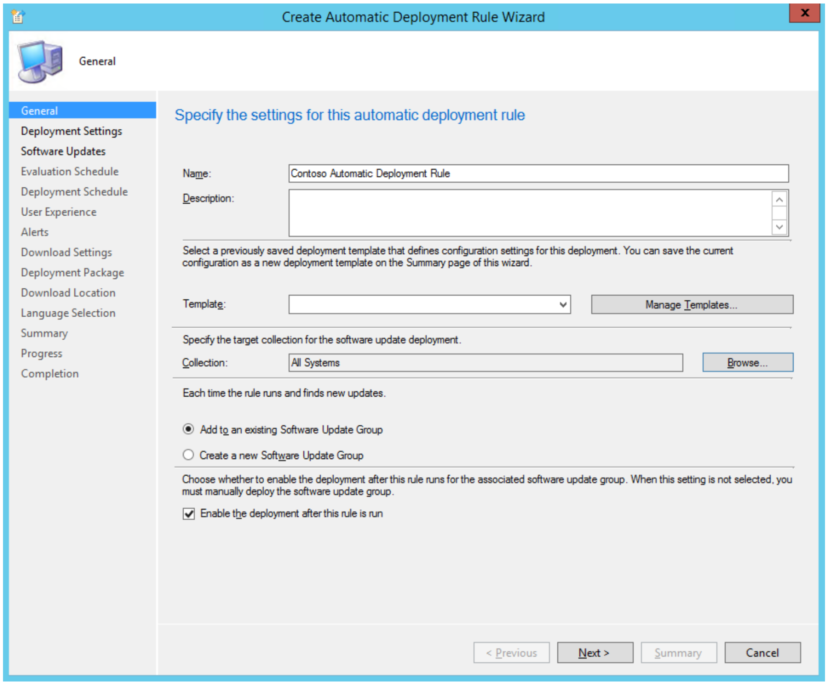
When an automatic deployment rule has been created and enabled, each time the rule runs, all software updates that meet the criteria defined within the rule are automatically downloaded from Microsoft, distributed to the assigned distribution points, and, finally, deployed to the collection specified in the automatic deployment rule. You can monitor or troubleshoot this activity by reviewing the RuleEngine.log on the primary site server.

NOTE The Systems Management Server (SMS) provider's computer account and the user that is running the wizard to download the software updates must both have Write NTFS permissions on the download location.

Deployments that are automatically created by automatic deployment rules are, by design, created only as required deployments. If you want to use the power of the automatic deployment rule capabilities for deploying updates, but also want to allow end users or server administrators the ability to apply the updates at their discretion, there are some alternative options to achieving close variations of this goal. One option is to set the deployment installation deadline defined within the rule for 12 months in the future. While this option affords end users or server owners flexibility in when they manually apply their updates, if no action is taken prior to the deadline, the potential for updates unexpectedly installing on these machines 12 months from the start time of the deployment is a caveat and therefore this option comes with some risk.

Another option is to clear the Enable The Deployment After This Rule Is Run check box on the General page of the wizard (see Figure 5-1), and use a Windows PowerShell script to create a deployment using any deployment options you choose.

The next step in creating a new automatic deployment rule is the Deployment Settings step (Figure 5-2). This step allows you to enable or disable the Wake-on-LAN option for required deployments, as well as the state message detail level for clients reporting back for each deployment created by the rule. Additionally, you can choose whether or not to automatically deploy software updates that include license agreements and whether or not to approve any license agreements.



Specify the settings for this automatic deployment rule

Name: Contoso Automatic Deployment Rule

Description:

Select a previously saved deployment template that defines configuration settings for this deployment. You can save the current configuration as a new deployment template on the Summary page of this wizard.

Template:

Specify the target collection for the software update deployment.

Collection: All Systems

Each time the rule runs and finds new updates.

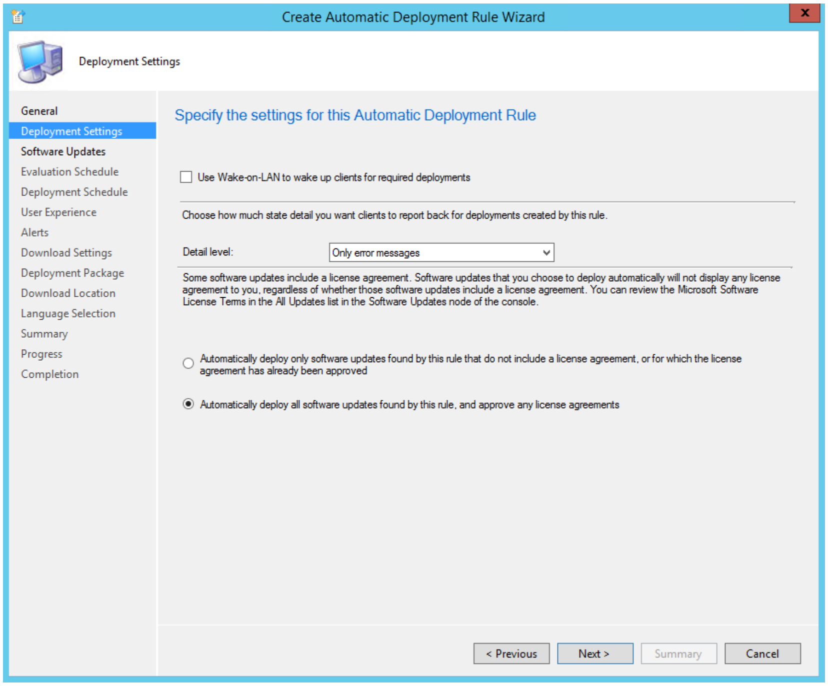
● Add to an existing Software Update Group

○ Create a new Software Update Group

Choose whether to enable the deployment after this rule runs for the associated software update group. When this setting is not selected, you must manually deploy the software update group.

☑ Enable the deployment after this rule is run

FIGURE 5-1 General page of the Create Automatic Deployment Rule Wizard



Specify the settings for this Automatic Deployment Rule

□ Use Wake-on-LAN to wake up clients for required deployments

Choose how much state detail you want clients to report back for deployments created by this rule.

Detail level: Only error messages

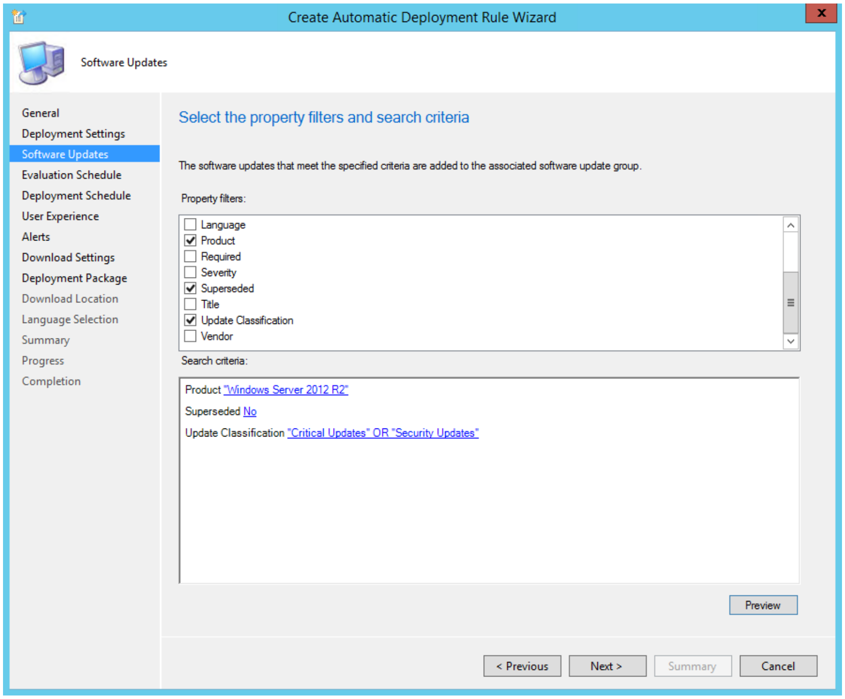
Some software updates include a license agreement. Software updates that you choose to deploy automatically will not display any license agreement to you, regardless of whether those software updates include a license agreement. You can review the Microsoft Software License Terms in the All Updates list in the Software Updates node of the console.

○ Automatically deploy only software updates found by this rule that do not include a license agreement, or for which the license agreement has already been approved

● Automatically deploy all software updates found by this rule, and approve any license agreements

FIGURE 5-2 Create Automatic Deployment Rule Wizard (Deployment Settings)

Next, as shown in Figure 5-3, select the individual property filters and search criteria for the updates to be automatically added to the associated software update group. One commonly used best practice is to always add "Superseded = No" to the search criteria to ensure there are no superseded updates included in the deployment.



Select the property filters and search criteria

The software updates that meet the specified criteria are added to the associated software update group.

Property filters:

- □ Language
- ☑ Product
- □ Required
- □ Severity
- ☑ Superseded
- □ Title
- ☑ Update Classification
- □ Vendor

Search criteria:

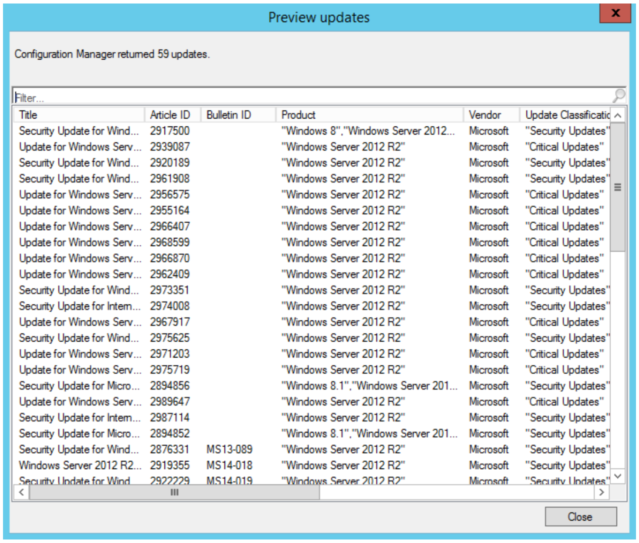
Product "Windows Server 2012 R2"

Superseded No

Update Classification "Critical Updates" OR "Security Updates"

FIGURE 5-3 Software Updates page of the Create Automatic Deployment Rule Wizard

Click Preview to display a list of the software updates that meet the defined search criteria, ensuring there are no undesired updates included that might require further filtering (see Figure 5-4).



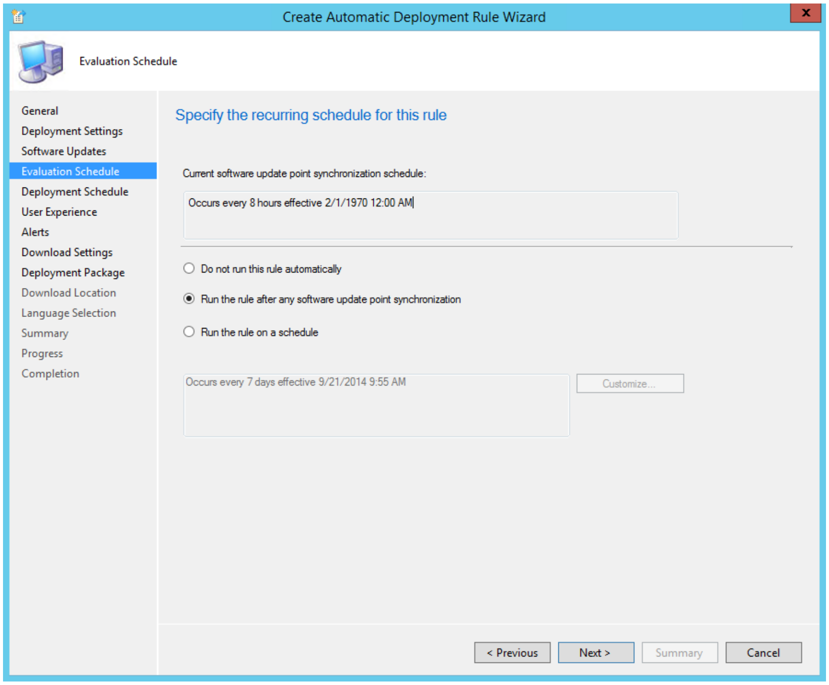
Configuration Manager returned 59 updates.

Title	Article ID	Bulletin ID	Product	Vendor	Update Classificatic
Security Update for Wind...	2917500		"Windows 8","Windows Server 2012...	Microsoft	"Security Updates"
Update for Windows Serv...	2939087		"Windows Server 2012 R2"	Microsoft	"Critical Updates"
Security Update for Wind...	2920189		"Windows Server 2012 R2"	Microsoft	"Security Updates"
Security Update for Wind...	2961908		"Windows Server 2012 R2"	Microsoft	"Security Updates"
Update for Windows Serv...	2956575		"Windows Server 2012 R2"	Microsoft	"Critical Updates"
Update for Windows Serv...	2955164		"Windows Server 2012 R2"	Microsoft	"Critical Updates"
Update for Windows Serv...	2966407		"Windows Server 2012 R2"	Microsoft	"Critical Updates"
Update for Windows Serv...	2968599		"Windows Server 2012 R2"	Microsoft	"Critical Updates"
Update for Windows Serv...	2966870		"Windows Server 2012 R2"	Microsoft	"Critical Updates"
Update for Windows Serv...	2962409		"Windows Server 2012 R2"	Microsoft	"Critical Updates"
Security Update for Wind...	2973351		"Windows Server 2012 R2"	Microsoft	"Security Updates"
Security Update for Intern...	2974008		"Windows Server 2012 R2"	Microsoft	"Security Updates"
Update for Windows Serv...	2967917		"Windows Server 2012 R2"	Microsoft	"Critical Updates"
Security Update for Wind...	2975625		"Windows Server 2012 R2"	Microsoft	"Security Updates"
Update for Windows Serv...	2971203		"Windows Server 2012 R2"	Microsoft	"Critical Updates"
Update for Windows Serv...	2975719		"Windows Server 2012 R2"	Microsoft	"Critical Updates"
Security Update for Micro...	2894856		"Windows 8.1","Windows Server 201...	Microsoft	"Security Updates"
Update for Windows Serv...	2989647		"Windows Server 2012 R2"	Microsoft	"Critical Updates"
Security Update for Intern...	2987114		"Windows Server 2012 R2"	Microsoft	"Security Updates"
Security Update for Micro...	2894852		"Windows 8.1","Windows Server 201...	Microsoft	"Security Updates"
Security Update for Wind...	2876331	MS13-089	"Windows Server 2012 R2"	Microsoft	"Security Updates"
Windows Server 2012 R2...	2919355	MS14-018	"Windows Server 2012 R2"	Microsoft	"Security Updates"
Security Update for Wind...	2922229	MS14-019	"Windows Server 2012 R2"	Microsoft	"Security Updates"

FIGURE 5-4 The Preview Updates dialog box of the Create Automatic Deployment Rule Wizard

The next step in the wizard is to define the evaluation schedule for the rule (see Figure 5-5). The Do Not Run This Rule Automatically option provides you the flexibility to manually run the rule at your discretion. The Run The Rule After Any Software Update Point Synchronization option triggers the rule to run each time the software update point completes a synchronization of updates from its defined synchronization source. This is commonly used for deploying regularly updated endpoint definition pattern files, but it is also useful for quickly deploying Patch Tuesday updates after their release to a pilot collection of test machines. The last option, Run The Rule On A Schedule, allows you to set a pre-defined recurring schedule for running the rule. This option allows you to configure the re-run pattern for the rule to never re-run, re-run only one time, as frequently as once per hour, or as infrequently as once per year. Additionally, you can click Customize to select a specific numeric calendar day, a specific weekday, the last day of the month, or from the first through the last weekday of each month for the scheduling criteria.

On the next page of the wizard (see Figure 5-6), you specify whether the deployment should be evaluated using Universal Coordinated Time (UTC) or the client's local time; when the software updates should be available and distributed to the content server; and the deadline for installing the required software updates.



Specify the recurring schedule for this rule

Current software update point synchronization schedule:

Occurs every 8 hours effective 2/1/1970 12:00 AM

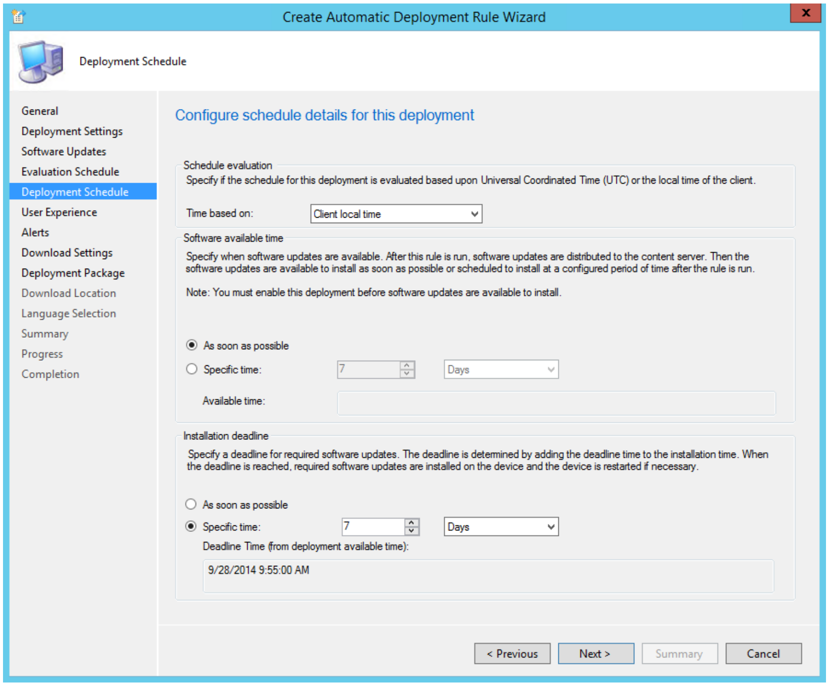
○ Do not run this rule automatically

● Run the rule after any software update point synchronization

○ Run the rule on a schedule

Occurs every 7 days effective 9/21/2014 9:55 AM

FIGURE 5-5 The Evaluation Schedule page of the Create Automatic Deployment Rule Wizard



Configure schedule details for this deployment

Schedule evaluation

Specify if the schedule for this deployment is evaluated based upon Universal Coordinated Time (UTC) or the local time of the client.

Time based on: Client local time

Software available time

Specify when software updates are available. After this rule is run, software updates are distributed to the content server. Then the software updates are available to install as soon as possible or scheduled to install at a configured period of time after the rule is run.

Note: You must enable this deployment before software updates are available to install.

● As soon as possible

○ Specific time: 7 Days

Available time:

Installation deadline

Specify a deadline for required software updates. The deadline is determined by adding the deadline time to the installation time. When the deadline is reached, required software updates are installed on the device and the device is restarted if necessary.

○ As soon as possible

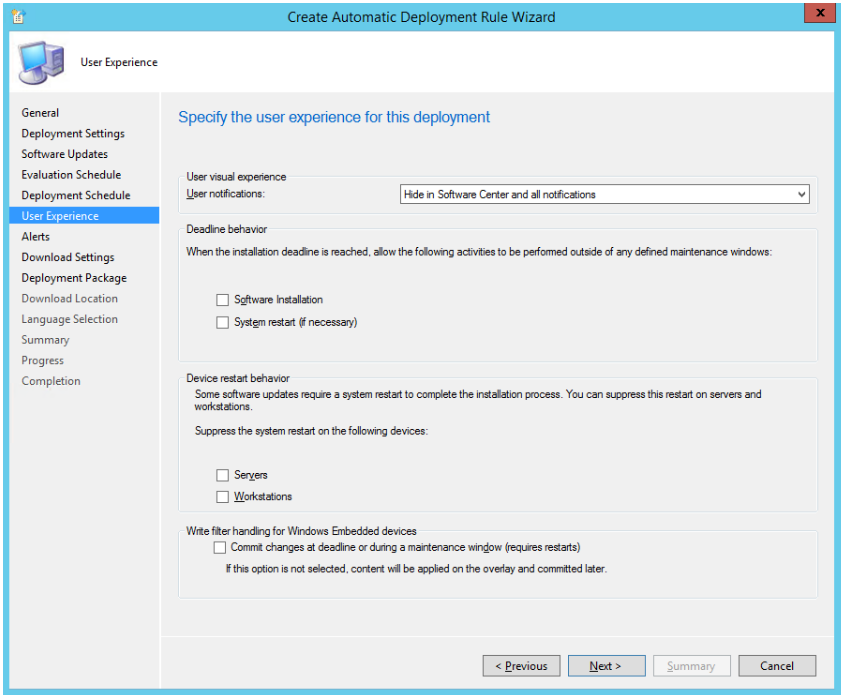
● Specific time: 7 Days

Deadline Time (from deployment available time):

9/28/2014 9:55:00 AM

FIGURE 5-6 The Deployment Schedule page of the Create Automatic Deployment Rule Wizard

On the next wizard page, shown in Figure 5-7, you can customize the user experience for the deployment by indicating whether to display or hide end-user notifications and by controlling restarts for servers and workstations both inside and outside of defined maintenance windows.



Specify the user experience for this deployment

User visual experience

User notifications: Hide in Software Center and all notifications

Deadline behavior

When the installation deadline is reached, allow the following activities to be performed outside of any defined maintenance windows:

☐ Software Installation

☐ System restart (if necessary)

Device restart behavior

Some software updates require a system restart to complete the installation process. You can suppress this restart on servers and workstations.

Suppress the system restart on the following devices:

☐ Servers

☐ Workstations

Write filter handling for Windows Embedded devices

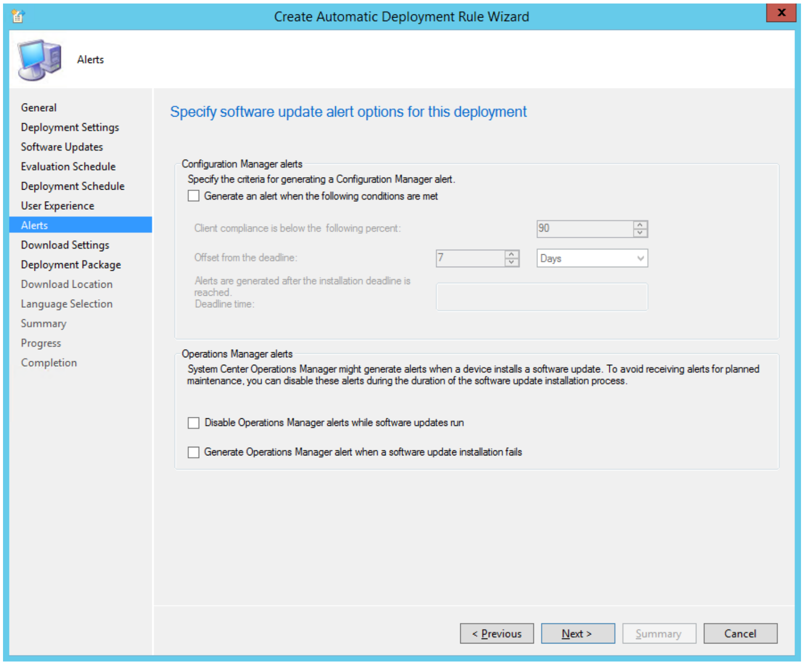
☐ Commit changes at deadline or during a maintenance window (requires restarts)

If this option is not selected, content will be applied on the overlay and committed later.

FIGURE 5-7 The User Experience page of the Create Automatic Deployment Rule Wizard

On the Alerts page of the wizard (see Figure 5-8), you can define alert options to leverage alerting capabilities internally through Configuration Manager or externally through System Center Operations Manager. You can configure Configuration Manager alerts using adjustable thresholds so that you can monitor deployment success rates.

On the next wizard page, shown in Figure 5-9, you can configure the software updates download behavior. Use the options on this page to control the content download behavior when clients are within a slow or unreliable boundary and when the software update content is not available on a preferred distribution point. Also on this page, you can select the check box to allow downloading of the update content from Microsoft Update Servers. Additionally, you can enable clients to share content with other clients in the same subnet. If this option is enabled, clients leverage the BranchCache feature to share software update content, which results in reduced bandwidth consumption across wide area networks. Finally, you can select the check box to control whether clients can download content while on a metered Internet connection in this step of the wizard.



Create Automatic Deployment Rule Wizard

Alerts

Specify software update alert options for this deployment

Configuration Manager alerts

Specify the criteria for generating a Configuration Manager alert.

□ Generate an alert when the following conditions are met

Client compliance is below the following percent: 90

Offset from the deadline: 7 Days

Alerts are generated after the installation deadline is reached.
Deadline time:

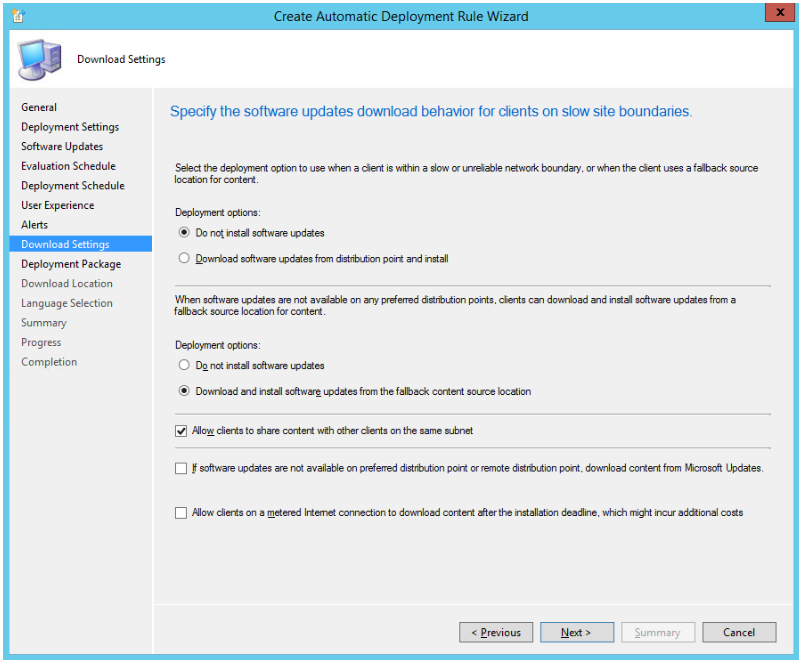
Operations Manager alerts

System Center Operations Manager might generate alerts when a device installs a software update. To avoid receiving alerts for planned maintenance, you can disable these alerts during the duration of the software update installation process.

□ Disable Operations Manager alerts while software updates run

□ Generate Operations Manager alert when a software update installation fails

FIGURE 5-8 The Software Update Alert Options page of the Create Automatic Deployment Rule Wizard



Create Automatic Deployment Rule Wizard

Download Settings

Specify the software updates download behavior for clients on slow site boundaries.

Select the deployment option to use when a client is within a slow or unreliable network boundary, or when the client uses a fallback source location for content.

Deployment options:

● Do not install software updates

○ Download software updates from distribution point and install

When software updates are not available on any preferred distribution points, clients can download and install software updates from a fallback source location for content.

Deployment options:

○ Do not install software updates

● Download and install software updates from the fallback content source location

☑ Allow clients to share content with other clients on the same subnet

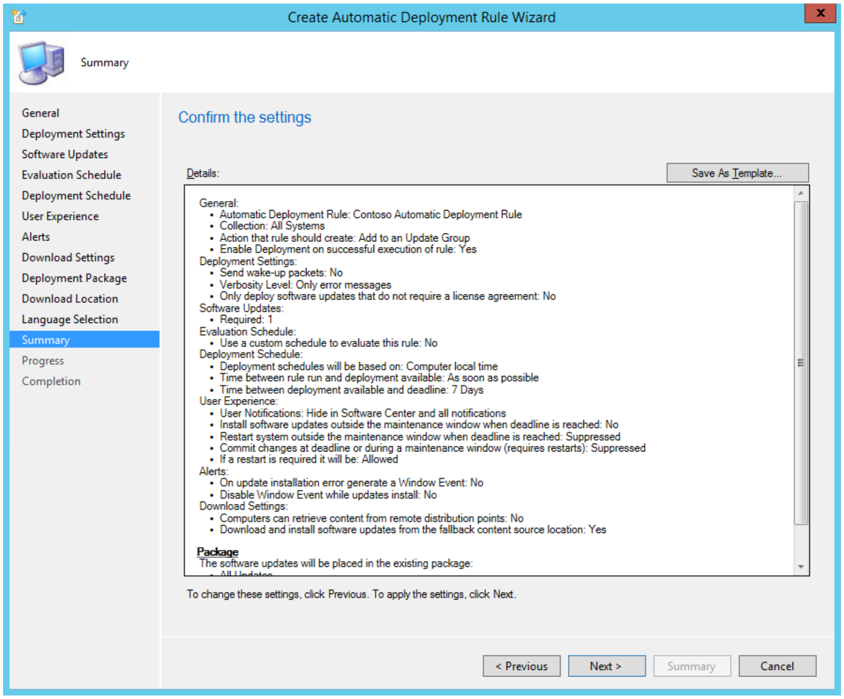
□ If software updates are not available on preferred distribution point or remote distribution point, download content from Microsoft Updates.

□ Allow clients on a metered Internet connection to download content after the installation deadline, which might incur additional costs

FIGURE 5-9 The Software Updates Download Behavior options page of the Create Automatic Deployment Rule Wizard

On the next three pages of the wizard, you can create a new software update package or add the updates to an existing package, indicate whether to download the updates from the Internet or from a location on the network, and select one or more applicable languages needed for the updates.

On the final page of the wizard, shown in Figure 5-10, you can save the selected settings to a template. You can use the saved template later when creating additional rules to avoid having to complete the full wizard when you want to use similar or the same options.



Confirm the settings

Details:

General:
- Automatic Deployment Rule: Contoso Automatic Deployment Rule
- Collection: All Systems
- Action that rule should create: Add to an Update Group
- Enable Deployment on successful execution of rule: Yes

Deployment Settings:
- Send wake-up packets: No
- Verbosity Level: Only error messages
- Only deploy software updates that do not require a license agreement: No

Software Updates:
- Required: 1

Evaluation Schedule:
- Use a custom schedule to evaluate this rule: No

Deployment Schedule:
- Deployment schedules will be based on: Computer local time
- Time between rule run and deployment available: As soon as possible
- Time between deployment available and deadline: 7 Days

User Experience:
- User Notifications: Hide in Software Center and all notifications
- Install software updates outside the maintenance window when deadline is reached: No
- Restart system outside the maintenance window when deadline is reached: Suppressed
- Commit changes at deadline or during a maintenance window (requires restarts): Suppressed
- If a restart is required it will be: Allowed

Alerts:
- On update installation error generate a Window Event: No
- Disable Window Event while updates install: No

Download Settings:
- Computers can retrieve content from remote distribution points: No
- Download and install software updates from the fallback content source location: Yes

Package
The software updates will be placed in the existing package:
- All Updates

To change these settings, click Previous. To apply the settings, click Next.

FIGURE 5-10 The Confirm the Settings page of the Create Automatic Deployment Rule Wizard

Automating software update database maintenance

Performing scheduled, routine maintenance on the software update point database is crucial to ensuring the ability to deploy software updates on time during the monthly patch deployment cycle. Keeping the WSUS database clean of expired metadata and regularly indexed will proactively avoid software update synchronization issues due to a fragmented or bloated database. Delivery of critical updates within the organization in a timely manner should be viewed as one of the highest priority tasks. In lieu of this, the framework on which software update deployments rely on should be maintained at a high priority.

It is recommended that you routinely perform WSUS cleanup actions on each software update point in your hierarchy. Whether you're using Windows Internal Database or a fullfeatured SQL Server database, it is recommended that you keep the Software Update Services (SUSDB) database of WSUS utilized by each software update point as clean of expired and superseded software update metadata as possible. An important procedural item to be aware of in this overall process is that if you have more than one software update point, you should run the cleanup process on the lowest tier servers first, then work your way up the hierarchy. If there are multiple software update points in the same tier, maintenance can be performed on them simultaneously. Additionally, selecting only the truly required update classifications and products for your environment is important for keeping the SUSDB database from becoming excessively large.

If an SUSDB database is hosted on Windows Internal Database, the operating system version affects how the WSUS cleanup maintenance tasks are performed. New Windows PowerShell cmdlets were introduced with the updated WSUS version that is included as a role in Windows Server 2012 R2. These new cmdlets enable SUSDB cleanup without the need for extensive custom scripts. Specifically, the Invoke-WsusServerCleanup cmdlet performs all of the WSUS maintenance tasks that historically have been accomplished either manually through the WSUS administrator console or via various community scripts. Windows Server versions prior to Windows Server 2012 R2 continue to require these scripts to initiate the WSUS maintenance tasks.

The Invoke-WsusServerCleanup cmdlet can be used to clean up obsolete computers, obsolete updates, unneeded content files, as well as declining, expired, and superseded updates. If WSUS is installed on an operating system version release earlier than Windows Server 2012 R2, you can use the following Windows PowerShell script to perform the WSUS metadata cleanup tasks:

```
[reflection.assembly]::LoadWithPartialName("Microsoft.UpdateServices.Administration")`
 | out-null
$wsus = [Microsoft.UpdateServices.Administration.AdminProxy]::GetUpdateServer();
$cleanupScope = new-object Microsoft.UpdateServices.Administration.CleanupScope;
$cleanupScope.DeclineSupersededUpdates = $true
$cleanupScope.DeclineExpiredUpdates = $true
$cleanupScope.CleanupObsoleteUpdates = $true
$cleanupScope.CompressUpdates = $true
$cleanupScope.CleanupObsoleteComputers = $true
$cleanupScope.CleanupUnneededContentFiles = $true
$cleanupManager = $wsus.GetCleanupManager();
$cleanupManager.PerformCleanup($cleanupScope);
```

In addition to the WSUS cleanup tasks, the SUSDB database should be regularly re-indexed to ensure optimal efficiency. A sample script that can be used to re-index the database on any version of WSUS is available at *https://gallery.technet.microsoft.com/scriptcenter/6f8cde49-5c52-4abd-9820-f1d270ddea61*. Similar to the WSUS cleanup tasks, the process of connecting to Windows Internal Database varies depending on which version of WSUS is installed. Windows Internal Database running on WSUS 3.0 and higher is based on SQL Server 2005 and can be managed using Microsoft SQL Server Management Studio Express, which can be downloaded from *http://www.microsoft.com/en-us/download/details.aspx?id=8961*, or by using the command-line tool SQLCMD.exe, which is provided within the Feature Pack for Microsoft SQL Server 2005, downloadable from *http://www.microsoft.com/en-us/download/details.aspx?id=15748*. A Windows Internal Database which is running on WSUS 4.0 and higher is based on SQL Server 2012 and can be managed using Microsoft SQL 2012 Server Management Studio Express, which can be downloaded from *http://www.microsoft.com/en-us/download/details.aspx?id=29062*, or by using SQLCMD.exe from the Microsoft SQL Server 2012 SP2 Feature Pack, which can be downloaded from *http://www.microsoft.com/en-us/download/details.aspx?id=43339*. The appropriate version of SQLCMD.exe can be run from a scheduled task to re-index the WSUS database at any frequency and time.

To connect to a SQL Server 2005 Windows Internal Database running on an operating system version of Windows Server 2008 R2 or earlier, use the following connection string from the command-line query tool SQLCMD.exe or use the graphical user interface in SQL Server Management Studio Express:

```
\\.\pipe\MSSQL$MICROSOFT##SSEE\sql\query
```

Below is a sample command line referencing the TechNet gallery script at *https://gallery.technet.microsoft.com/scriptcenter/6f8cde49-5c52-4abd-9820-f1d270ddea61*, which can be run from a scheduled task. It can be used to run the TechNet gallery maintenance script against an SUSDB running on a SQL Server 2005 or SQL Server 2008 Windows Internal Database:

```
SQLCMD.exe -S \\.\pipe\MSSQL$MICROSOFT##SSEE\sql\query -i
"<scriptlocation>\WsusDBMaintenance.sql"
```

To connect to a Windows Internal Database based on SQL Server 2012 and run any individual SQL queries or maintenance scripts, replace the preceding connection string with the following connection string, which can also be run using SQLCMD.exe provided in the SQL Server 2012 SP2 Feature Pack or SQL Server 2012 Management Studio Express:

```
\\.\pipe\MICROSOFT##WID\tsql\query
```

The following sample command line can be run from a scheduled task to run the same TechNet gallery maintenance script, which can be run against a SQL Server 2012 Windows Internal Database:

```
SQLCMD.exe -S \\.\pipe\Microsoft##WID\tsql\query -i
"<scriptlocation\WsusDBMaintenance.sql"
```

Performing Windows Internal Database backups on the SUSDB database is another important task that should occur on a nightly recurring schedule. Similar to the SUSDB database indexing process, regular database backups of the SUSDB database hosted on Windows Internal Database can be accomplished using a scheduled task that runs SQLCMD.exe with the appropriate switches. The following sample script can be saved with a

.SQL extension and run from SQLCMD.exe. This example query creates a full backup of the SUSDB database and saves it to a path of C:\Backup\Susdb.bak.

```
BACKUP DATABASE [SUSDB] TO DISK = N'c:\backup\susdb.bak' WITH NOFORMAT, NOINIT, NAME =
N'SUSDB-Full Database Backup', SKIP, NOREWIND, NOUNLOAD, STATS = 10
GO
```

Site server software update automation

Familiarizing yourself with the various WMI classes that exist on a primary site server that can be utilized for automating software update related tasks is helpful for things like automating monthly software update group maintenance, as well as performing cleanup actions to keep software update groups clean. This section covers the basics for communicating with the WMI provider on a site server, and using the results and WMI methods to automatically create, delete, or update new or existing software update groups and update packages.

A variety of tools can help find the applicable WMI namespaces, classes, and methods that are applicable to automate any processes or tasks that you do manually. Assume your primary site code is ABC. To query the provider on your site server, you would use the WMI namespace root\sms\site_abc replacing the *abc* with the actual site code of the primary site you want to run the query against. One popular tool for browsing and viewing WMI namespaces, classes, instances, and properties is WMI Explorer, which can be downloaded from CodePlex at *http://wmie.codeplex.com*.

Before diving into the various Configuration Manager WMI namespaces and classes, it is helpful to understand some of the acronyms used when working with software updates. CI is short for *configuration item* and can contain one or more elements along with their validation criteria. A CI can be thought of as a discrete unit of configuration to assess for compliance. An SDM package is short for *service definition model package*. An SDM package is sometimes described as using SML, or *service modelling language*. SDM packages are connected with CIs and provide further information about what configuration is affected during the evaluation or implementation of various software update activities.

The first WMI class covered is SMS_SoftwareUpdate. This WMI class contains all of the software updates that are visible in the All Software Updates node of the Configuration Manager console. Each software update has its own unique property named CI_ID and CI_UniqueID, which are properties that can be used when referencing a specific software update within an automation script. It is helpful to add these two columns to your Configuration Manager console so that you can more easily refer to these values when testing or when automating a given task against a given update or updates. The following sample Windows PowerShell script creates a new software update group named Win7_Critical_Updates. To accomplish this task, two mandatory parameters, -SiteCode and -SiteServer, must be provided either in the Windows PowerShell command line following the script file name or when prompted after running the script. These two parameters make the script portable, meaning they can run against any site code and target any site server. The site server's WMI provider is first queried for all software updates. The results are then filtered to identify only the updates that apply to the Windows 7 operating system and which are of a critical severity level. The results are returned as an array and set to a parameter named $UpdateList. Each member of this array is then processed using a Foreach loop, which extracts the CI_ID from each software update to build another array named $UL. Finally, the Set-Location cmdlet sets the location to the primary site server name specified in the -SiteServer parameter and the site code specified in the -SiteCode parameter. Finally, the New-CMSoftwareUpdateGroup cmdlet is run to actually create the software update group. The New-CMSoftwareUpdateGroup cmdlet includes the -Name parameter, which is used to populate the name of the software update group, and the -Description parameter, which is used to populate the description of the new software update group. The parameter $UL is an array containing each of the CI_IDs returned by the Foreach loop. This array is provided to the -UpdateID parameter so that the applicable software updates that meet the filtered criteria are automatically added to the software update group.

```
Param(
        [Parameter(Mandatory = $true)]
        $SiteCode,
        [Parameter(Mandatory = $true)]
        $SiteServer
     )

Import-Module ($Env:SMS_ADMIN_UI_PATH.Substring(0,$Env:SMS_ADMIN_UI_PATH.Length-5) +
'\ConfigurationManager.psd1')

$UL=@()
$UpdateList = @(Get-WmiObject -NameSpace root\SMS\Site_$($SiteCode) -ComputerName
$SiteServer -Query "Select * from SMS_SoftwareUpdate Where SeverityName='Critical' and
LocalizedCategoryInstanceNames='Windows 7' and IsExpired='False'")
Foreach ($Update in $UpdateList)
{
$UL+=$Update.CI_ID
}
CD "$($SiteCode):"
New-CMSoftwareUpdateGroup -Name "Win7_Critical_Updates" -Description "Windows 7 Critical
Updates" -UpdateId $UL
```

Whether you create a software update group manually with the administrator console or by using a script, you can review the results from running the above script by reviewing the SMSProv.log on the primary site server. An instance is created in the SMS_AuthorizationList WMI class, which is an SMS provider server class that contains a collection of SMS_SoftwareUpdate objects for the software updates available on the site and authorized for deployment. If you have an existing software update group to add new software updates to, you can use the Add-CMSoftwareUpdateToGroup cmdlet, as shown in the following example:

```
$Criticals = Get-WmiObject -NameSpace "root\SMS\Site_PRI" -Class "SMS_SoftwareUpdate" -
Filter "SeverityName='Critical' and LocalizedCategoryInstanceNames='Windows 7'"
```

```
foreach ($Update in $criticals)
{
Add-CMSoftwareUpdateToGroup -SoftwareUpdateGroupName "MyUpdateGroup" -SoftwareUpdateID
$Update.CI_ID
}
```

The following Windows PowerShell script sample creates a maintenance window based on a specified number of days following Patch Tuesday for a specified length in hours, and applies it to a specified collection. The SecondTuesday function used in this example can be re-used for any other scripts that require programmatically determining which numeric day Patch Tuesday falls on during the current month. The maintenance window created in this sample is also configured to apply only to software updates.

```
Param(
        [Parameter(Mandatory=$true,
        HelpMessage="CollectionID?",Position=1)]
        [String]
        [ValidateNotNullOrEmpty()]
        $CollectionID = $(Throw "CollectionID is Required"),
        [Parameter(Mandatory=$true,
        HelpMessage="Hour of day between 0-23 to begin?",Position=2)]
        [INT]
        [ValidateNotNullOrEmpty()]
        $MilitaryStartHour = $(Throw "Hour of day between 0-23 to begin"),
        [Parameter(Mandatory=$true,
        HelpMessage="Length in hours of Maintenance Window?",Position=3)]
        [INT]
        [ValidateNotNullOrEmpty()]
        $WindowLengthInHours= $(Throw "Length in hours of Maintenance Window"),
        [Parameter(Mandatory=$true,
        HelpMessage="Days after Patch Tuesday to begin?",Position=4)]
        [INT]
        [ValidateNotNullOrEmpty()]
        $NumDaysAfterPatchTuesday = $(Throw "Number of days after Patch Tuesday to begin
is required")

)

#Get directory the script is running in
function Get-ScriptDirectory
{
    $Invocation = (Get-Variable MyInvocation -Scope 1).Value;
    if($Invocation.PSScriptRoot)
    {
        $Invocation.PSScriptRoot;
    }
    Elseif($Invocation.MyCommand.Path)
    {
        Split-Path $Invocation.MyCommand.Path
    }
    else
    {
$Invocation.InvocationName.Substring(0,$Invocation.InvocationName.LastIndexOf("\"));
    }
}



#Import ConfigMgr Module and CD to the site
 Import-Module ($Env:SMS_ADMIN_UI_PATH.Substring(0,$Env:SMS_ADMIN_UI_PATH.Length-5) +
'\ConfigurationManager.psd1')
 $PSD = Get-PSDrive -PSProvider CMSite
 CD "$($PSD):"

#Delete all existing maintenance windows
$WindowNames = Get-CMMaintenanceWindow -CollectionID $CollectionID | Select Name
foreach ($Window in $WindowNames)
{Remove-CMMaintenanceWindow -Name $Window.Name -CollectionID $CollectionID -Force}

#Figure out the date
$Date = Get-Date -DisplayHint Date
$TodayMonth = $Date.Month
$TodayYear = $Date.Year

#Returns a date object of the second tuesday of the month for given month and year
 function SecondTuesday ([int]$Month, [int]$Year) {
     [int]$Day = 1
     while((Get-Date -Day $Day -Hour 0 -Millisecond 0 -Minute 0 -Month $Month -Year
$Year -Second 0).DayOfWeek -ne "Tuesday") {
         $day++
     }
     $day += 7
     return (Get-Date -Day $Day -Hour 0 -Millisecond 0 -Minute 0 -Month $Month -Year
$Year -Second 0)
 }

$SecTues = SecondTuesday $TodayMonth $TodayYear

 #Add appropriate number of days following Patch Tuesday to begin and set start and end
time of maintenance window
```

```
 $StartDate = $SecTues.AddDays($NumDaysAfterPatchTuesday)
 $StartTime = $StartDate.AddHours($MilitaryStartHour)
 $EndTime = $StartTime.AddHours($WindowLengthInHours)

 #Dynamically name the maintenance window using the maintenance window start date and
time
 $MaintWindowName = $StartDate.ToShortDateString() + " " +
$StartTime.ToShortTimeString()

 #Create The ScheduleToken
 $Schedule = New-CMSchedule -Start $StartTime -End $EndTime -NonRecurring

 #Create the Maintenance Window on the appropriate collection
 New-CMMaintenanceWindow -CollectionID $CollectionID -ApplyToSoftwareUpdateOnly -Name
$MaintWindowName -Schedule $Schedule
```

One of the caveats regarding using the automatic deployment rule capability is that you are not able to utilize an automatic deployment rule to create an available deployment; automatic deployment rules are designed to create only required deployments. The following sample script creates a customizable and automatic deployment based on an existing software update group that was pre-created by an automatic deployment rule. It automatically determines the exact date Patch Tuesday falls on during the current month and also allows you to define the number of days following Patch Tuesday to make the deployment available to clients. Customizing the parameters on the Start-CMSoftwareUpdateDeployment line at the end of the following script allows you to create the deployment using any number of deployment options by modifying the parameters, including deploying a software update group created with an automatic deployment rule as an available deployment rather than as a required one.

```
<#
Disclaimer:
This Sample Code is provided for the purpose of illustration only and is not intended to
be used in a production environment. THIS SAMPLE CODE AND ANY RELATED INFORMATION ARE
PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING
BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A
PARTICULAR PURPOSE.  We grant You a nonexclusive, royalty-free right to use and modify
the Sample Code and to reproduce and distribute the object code form of the Sample Code,
provided that You agree: (i) to not use Our name, logo, or trademarks to market Your
software product in which the Sample Code is embedded; (ii) to include a valid copyright
notice on Your software product in which the Sample Code is embedded; and (iii) to
indemnify, hold harmless, and defend Us and Our suppliers from and against any claims or
lawsuits, including attorneys' fees, that arise or result from the use or distribution
of the Sample Code.

.SYNOPSIS

This script can be used to assist with automatically deploying an Automatic Deployment
```

```
Rule created Software Update Group based on a specified number of days after Patch
Tuesday for the current month.

.DESCRIPTION

This script will do the following:
1) Prompt for an existing Software Update Group Name
2) Prompt for an existing Collection Name
3) Prompt for a new deployment name and and a description to give to the deployment
4) Prompt for the number of days after "Patch Tuesday" to make the deployment available
5) Prompt if you want the deployment "Available" or "Required"
6) Prompt whether you want the deployment to be available in "UTC" or "LocalTime"

.PARAMETER UpdateGroupName

UpdateGroupName is the name of the existing Software Update Group to deploy

.PARAMETER CollectionName

CollectionName is the name of the existing collection to target the deployment

.PARAMETER DeploymentName

DeploymentName is the name to give to the new deployment that will be created

.PARAMETER DeploymentDescription

DeploymentDescription is the description for the new deployment that will be created

.PARAMETER AddDays

AddDays is the number of days following Patch Tuesday to make the deployment available
to clients

.PARAMETER DepType

DepType is used to specify whether to make the deployment "Available" or "Required"

.PARAMETER Time

Specify either 'LocalTime' or 'UTC' depending on the requirement

.EXAMPLE

.\PatchTuesdayCreator.ps1 -UpdateGroupName MyUpdateGroup -CollectionName "All Systems" -
DeploymentName "My Deployment" -DeploymentDescription "My Patch Tuesday Deployment" -
```

```
AddDays 7 -DepType Available -Time LocalTime
#>

Param(

        [Parameter(Mandatory=$true,
               HelpMessage="Please enter the Software Update Group Name",Position=0)]
        [String]
        [ValidateNotNullOrEmpty()]
        $UpdateGroupName = $(Throw "Software Update Group Required"),



        [Parameter(Mandatory=$true,
               HelpMessage="Please enter the Collection Name",Position=1)]
        [String]
        [ValidateNotNullOrEmpty()]
        $CollectionName = $(Throw "Collection Name is Required"),

        [Parameter(Mandatory=$true,
               HelpMessage="What would you like to name the deployment?",Position=2)]
        [String]
        [ValidateNotNullOrEmpty()]
        $DeploymentName = $(Throw "Deployment Name is Required"),

        [Parameter(Mandatory=$true,
               HelpMessage="What would you like for the Deployment
Description?",Position=3)]
        [String]
        [ValidateNotNullOrEmpty()]
        $DeploymentDescription = $(Throw "Deployment Description is Required"),

        [Parameter(Mandatory=$true,
         HelpMessage="What would you like for the Deployment Description?",Position=4)]
        [INT]
        [ValidateNotNullOrEmpty()]
        $AddDays = $(Throw "Number of Days following Patch Tuesday to make the
deployment available?"),

                 [Parameter(Mandatory=$true,
         HelpMessage="Make the deployment available or required?",Position=5)]
        [String]
        [ValidateNotNullOrEmpty()]
        $DepType = $(Throw "Make the deployment Available or Required?"),

                 [Parameter(Mandatory=$true,
         HelpMessage="LocalTime or UTC?",Position=6)]
        [String]
        [ValidateNotNullOrEmpty()]
        $Time = $(Throw "LocalTime or UTC?")
)

#Get today's date
$Date = Get-Date

#Convert it to MM/DD/YYYY format
$ShortDate = $Date.ToShortDateString()
$AvailableDate = ($Date.AddDays($AddDays)).ToShortDateString()

#This function returns a date object of the second Tuesday of the month for the current
month
 function SecondTuesday ([int]$Month, [int]$Year)
 {
     [int]$Day = 1
     while((Get-Date -Day $Day -Hour 0 -Millisecond 0 -Minute 0 -Month $Month -Year
$Year -Second 0).DayOfWeek -ne "Tuesday")
     {
         $day++
     }
        $day += 7
        Return (Get-Date -Day $Day -Hour 0 -Millisecond 0 -Minute 0 -Month $Month -Year
$Year -Second 0)
 }

#Determine the second Tuesday of the current month
$SecTues = (SecondTuesday $Date.Month $Date.Year).ToShortDateString()

#Make sure the second Tuesday hasn't already passed
If ((Get-Date $ShortDate) -le (Get-Date $SecTues))
{
    #Load ConfigMgr Powershell Module
    Import-Module ($Env:SMS_ADMIN_UI_PATH.Substring(0,$Env:SMS_ADMIN_UI_PATH.Length-5) +
'\ConfigurationManager.psd1')
    #Determine site code
    $PSD = Get-PSDrive -PSProvider CMSite
    #Change into ConfigMgr site namespace
    Set-Location "$($PSD):"
Try
    {
    #Create the deployment
    Start-CMSoftwareUpdateDeployment -SoftwareUpdateGroupName $UpdateGroupName -
CollectionName $CollectionName -DeploymentName $DeploymentName -Description
$DeploymentDescription -DeploymentType $DepType -VerbosityLevel
OnlySuccessAndErrorMessages -TimeBasedOn $Time -DeploymentAvailableDay $AvailableDate -
UserNotification DisplayAll -PersistOnWriteFilterDevice $false -
DisableOperationsManagerAlert $true -GenerateOperationsManagerAlert $false -
ProtectedType RemoteDistributionPoint -UnprotectedType UnprotectedDistributionPoint -
UseBranchCache $true -DownloadFromMicrosoftUpdate $true
    Write-Host "Deployment $DeploymentName Created against Collection Name
$CollectionName for an available date of $AvailableDate"
    }
Catch
    {
    Write-Host "$($_.Exception.Message) Please confirm you are using an existing
Software Update Group, and providing a valid Collection name"
    }

}
Else {Write-Host "Patch Tuesday has already passed.  Please run prior to Patch
Tuesday."}
```

Client software update automation

Familiarizing yourself with the various WMI classes and namespaces that exist on Configuration Manager installed clients that are utilized for software updates is helpful for automating a variety of software update-related administrative tasks. Some examples of such tasks are additional software updates actions following the application of a new image to a client or a step during a System Center Orchestrator runbook. Having a thorough knowledge of the various WMI classes that are applicable to the software update capability can help you better understand what's required to implement a variety of software update-related actions. Also, it is helpful to know that software updates are referenced using a unique update identifier in the software update log files and that you can view the GUID of each update by adding the Unique Update ID column to your Configuration Manager console when viewing software updates.

A Windows PowerShell command line such as the one in the following example can be used to determine which updates have been deployed but are not yet installed on the client. The first step is to determine which updates are non-compliant. The WMI class CCM_SoftwareUpdate in the WMI namespace root\CCM\ClientSDK provides all software updates that are applicable but not yet installed. This namespace can be accessed remotely on clients using Windows PowerShell Remote Management (WinRM). Below is a simple Windows PowerShell command to display a grid of all the updates that are applicable to the client:

```
Get-WMIObject -Class CCM_SoftwareUpdate -Namespace root\CCM\ClientSDK | Select
ArticleID, Description | Out-Gridview
```

The list of non-compliant software updates can in turn be provided by invoking the WMI method InstallUpdates within the WMI class CCM_SoftwareUpdatesManager to initiate the installation of all non-compliant updates.

The first line of the following script retrieves all instances of CCM_SoftwareUpdate from the root\CCM\ClientSDK WMI namespace. The second line retrieves the missing updates and turns the single object into an array of WMI objects. Finally, the third line invokes the InstallUpdates method in the CCM_SoftwareUpdate class and passes a flat array as the -ArgumentList parameter.

```
$MissingUpdates = Get-WMIObject -Namespace root\CCM\ClientSDK -Class CCM_SoftwareUpdate
-Filter ComplianceState=0
$MissingUpdatesReformatted = @(MissingUpdates | Foreach-Object {if($_.ComplianceState -
eq 0){[WMI]$_.__PATH}})
$InstallReturn = Invoke-WMIMethod -Namespace root\CCM\ClientSDK -Class
CCM_SoftwareUpdatesManager -Name InstallUpdates -ArgumentList
(,$MissingUpdatesReformatted)
```

The WMI class CCM_AssignmentCompliance in the WMI namespace root\CCM\SoftwareUpdates\DeploymentAgent provides all software update assignments along with the associated assignment GUIDs for each software update. Running a Windows PowerShell command such as the following example displays all software update assignments targeted to the client and whether each assignment is compliant.

```
Get-WMIObject -Namespace root\ccm\SoftwareUpdates\DeploymentAgent -Class
CCM_AssignmentCompliance | Select AssignmentID, IsCompliant
```

The WMI class CCM_UpdateStatus in the namespace root\CCM\SoftwareUpdates\UpdatesStore contains the most recent status for all updates on a client. Running a Windows PowerShell command such as the following example displays all updates on the client and whether they are installed or missing.

```
Get-WMIObject -Namespace root\ccm\SoftwareUpdates\UpdatesStore -Class CCM_UpdateStatus |
Select Title, Status
```

Along with the WMI Explorer tool, another tool that is helpful for accessing the client side WMI namespaces remotely is Client Center for Configuration Manager, downloadable from *http://sccmclictr.codeplex.com*. This tool is also helpful for learning the various Windows PowerShell classes, namespaces, and commands to access client side software updates information, since it displays the Windows PowerShell commands being run against the client for each action, including various software update actions.

Community resources for software update automation

A variety of blogs offer software update automation scripts and information:

- Steve Rachui's blog

  http://blogs.msdn.com/b/steverac/archive/2014/06/12/automating-software-updates.aspx

- Jason Githens's blog

  http://technet.microsoft.com/en-us/video/automating-deployments-of-software-updates.aspx

- Deepak Singh Dhami's blog

  http://www.dexterposh.com/2014/06/powershell-sccm-2012-automate-patching.html

- David O'Brien's blog

  http://www.david-obrien.net/

In addition, see the following documentation on MSDN:

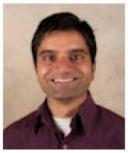
- *http://msdn.microsoft.com/en-us/library/hh949569.aspx*
- *http://msdn.microsoft.com/en-us/library/jj217901.aspx*

About the authors

ANDRE DELLA MONICA is a Premier Field Engineer for Microsoft and has been working with System Center Configuration Manager since it was known as SMS. Before becoming a Premier Field Engineer, he was recognized as a top Support Engineer on Consumer Technical Support for Microsoft Platform products. Andre attended college at Sao Paulo, Brazil, and earned his technology degree in Computer Network Management. He resides in Houston, Texas, with Rose his wife and his daughter Sarah. In his free time, he enjoys producing and recording music, as well as being an Xbox gamer.

CHRIS SHILT is a Premier Field Engineer with Microsoft and has been working with System Center Configuration Manager, in one iteration or another, for over 12 years. Prior to working at Microsoft, he managed software updates for a major command with the U.S. Air Force. A lifelong resident of Ohio, he lives in a small town north of Dayton with Annie, his wife of 21 years. He is the proud father of Zach and Jessie, both of whom will graduate from college this spring.

RUSS RIMMERMAN is a Microsoft Premier Field Engineer with a primary focus on System Center Configuration Manager. During his 12-year tenure working with Configuration Manager, he has engaged with customers with moderately simple to ultra-complex Configuration Manager hierarchies of all sizes in nearly every industry. Before joining Microsoft, Russ spent five years in the U.S. Air Force and also spent time as both a consultant and as a senior IT administrator. He currently resides in Houston, Texas, and in his free time he enjoys swimming, anything involving computers, and, most importantly, spending time with his wife and new twin boys, Owyn and Jaxen.



RUSHI FALDU, a Senior Premier Field Engineer supporting System Center Configuration Manager and Microsoft Intune, has been with Microsoft for over nine years. He has worked with the product since it was known as SMS 2.0. He is a Technology Manager for Configuration Manager and Microsoft Intune products. He is also a lead for the System Center Concepts and Administration and Enterprise Mobility Suite: Managing Devices with InTune and Configuration Manager workshops. He was lead author for the ebooks *Microsoft System Center: Troubleshooting Configuration Manager* and *Microsoft System Center: Configuration Manager Field Experience* from Microsoft Press. Rushi resides in New Jersey and enjoys P90X workouts and running outside in his free time. He loves hiking, camping, and playing tennis with his daughters.

(CONTRIBUTING AUTHOR) SAMEER PATIL, a consultant for Microsoft Global Delivery, has been working with System Center Configuration Manager since 2007. He has been working, globally, on various deployment projects that use Configuration Manager as a key infrastructure product in the solutions. Sameer attended college at Navi Mumbai, India, earning an engineering degree. He resides in Mumbai, India, and travels for work most of the time.

About the series editor

MITCH TULLOCH is a well-known expert on Windows Server administration and cloud computing technologies. He has published hundreds of articles on a wide variety of technology sites and has written, contributed to or been series editor for over 50 books. Mitch is one of the most popular authors at Microsoft Press—the almost two dozen ebooks on Windows Server and System Center he either wrote or was Series Editor on have been downloaded more than 2.5 million times! For a complete list of free ebooks from Microsoft Press, visit the Microsoft Virtual Academy at *http://www.microsoftvirtualacademy.com/ebooks*.

Mitch has repeatedly received Microsoft's Most Valuable Professional (MVP) award for his outstanding contributions to supporting the global IT community. He is a ten-time MVP in the technology area of Windows Server Software Packaging, Deployment & Servicing. You can find his MVP Profile page at *http://mvp.microsoft.com/en-us/mvp/Mitch%20Tulloch-21182*.

Mitch is also Senior Editor of WServerNews, a weekly newsletter focused on system admin and security issues for the Windows Server platform. With almost 100,000 IT pro subscribers worldwide, WServerNews is the most popular Windows Server–focused newsletter in the world. Visit *http://www.wservernews.com* and subscribe to WServerNews today!

Mitch also runs an IT content development business based in Winnipeg, Canada, that produces white papers and other collateral for the business decision maker (BDM) and technical decision maker (TDM) audiences. His published content ranges from white papers about Microsoft cloud technologies to reviews of third-party products designed for the Windows Server platform. Before starting his own business in 1998, Mitch worked as a Microsoft Certified Trainer (MCT) for Productivity Point.

For more information about Mitch, visit his website at *http://www.mtit.com*. You can also follow Mitch on Twitter @mitchtulloch.

Free ebooks